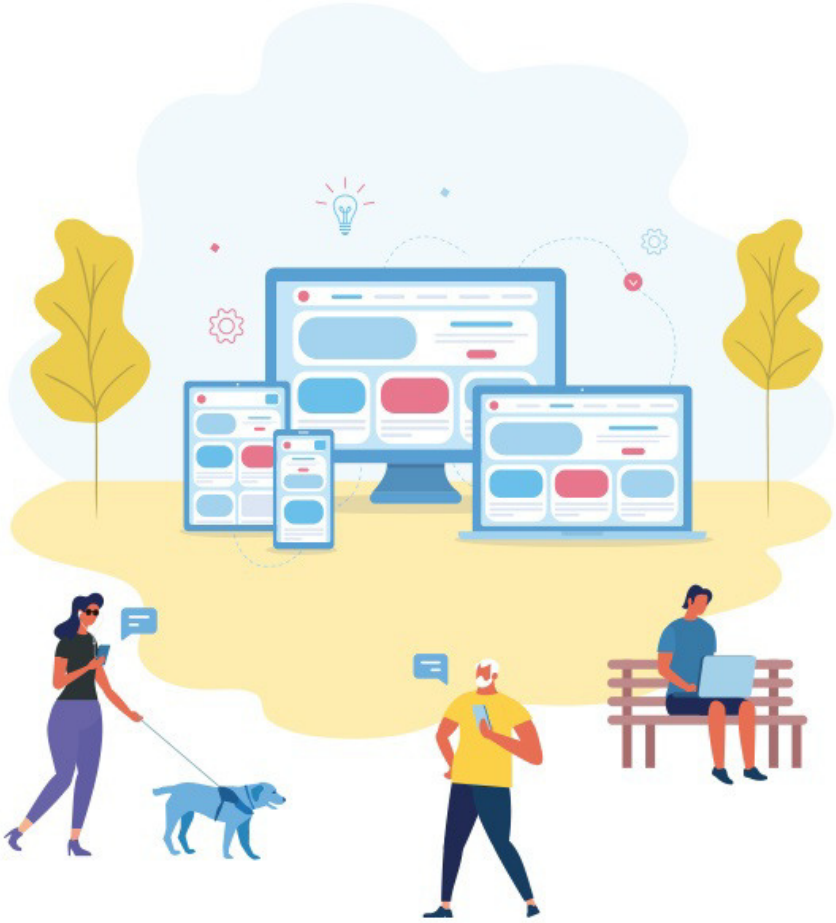
Acessibilidade na Web

Boas práticas para construir sites e aplicações acessíveis



Casa do Código

REINALDO FERRAZ

ISBN

Impresso e PDF: 978-65-86110-10-4

EPUB: 978-65-86110-16-6

MOBI: 978-65-86110-11-1

Caso você deseje submeter alguma errata ou sugestão, acesse http://erratas.casadocodigo.com.br.

PREFÁCIO

POR LÊDA LÚCIA SPELTA[1]

Ao escrever esta apresentação, meu real desafio é o de ser sintética. Tenho estado envolvida profissionalmente com a questão da acessibilidade há quase vinte anos. Digo profissionalmente porque, na vida prática, este envolvimento começou provavelmente quando pela primeira vez tentei procurar algum objeto dentro do berço, já que nasci com baixíssima visão, que fui perdendo ao longo da vida. Na virada do século XXI, a necessidade de uma web acessível para manter a minha independência, tanto nas atividades pessoais quanto na profissão de analista de sistemas, se aliou à curiosidade técnica pelas diretrizes de acessibilidade que acabavam de ser publicadas pelo W3C. Eu sabia bem que a quase totalidade das pessoas não fazia a menor ideia de que usuários com diferentes habilidades acessavam a web de maneiras diferentes e que existiam técnicas para criar páginas web que tornavam possíveis esses acessos. E assim, comecei a pensar que poderia escrever algo que ajudasse nessa divulgação. Minha inserção na acessibilidade tem, portanto, três raízes: a usuária, a técnica e a motivadora.

Conheci o Reinaldo Ferraz quando ele começou a se interessar pela acessibilidade. E aqui também caberia dizer: se interessar profissionalmente; pois nenhuma ação de acessibilidade tem chance de se manter ao longo do tempo, se não houver um desejo genuíno de incluir as pessoas. E a convivência com o Reinaldo me mostrou que este desejo, ele também deve ter trazido do berço,

pois o pensamento inclusivo é algo que transparece naturalmente em todas as suas atitudes, da conferência ao chope, das declarações oficiais às conversas informais... E assim, com sua aguda inteligência e grande sensibilidade, aliadas ao trabalho árduo e dedicado, em poucos anos conquistou uma posição de destaque neste assunto. Além do perfil técnico, a sua atuação profissional como responsável pelas ações de acessibilidade do Escritório Brasil do W3C (órgão internacional para o impulsionamento e padronização da web) lhe proporciona uma visão panorâmica do estado atual das diretrizes e ações de acessibilidade, em nível nacional e internacional.

Assim como a acessibilidade na web destina-se a um público de usuários tão amplo e diversificado que chega a englobar todas as pessoas, também o próprio livro está escrito de forma muito acessível e se destina a um público bastante abrangente. Qualquer profissional ou usuário interessado em acessibilidade na web, mesmo que não tenha conhecimento prévio, encontrará aqui os conceitos básicos, as especificidades relativas aos seus diversos públicos, uma introdução sobre as várias diretrizes e leis existentes, bem como orientações básicas para a sua implementação, manutenção, avaliação, correção e divulgação.

Porém, apesar de atender a este público amplo, enganam-se os que pensarem tratar-se de uma obra introdutória. Seu escopo abrange desde conceitos fundamentais e orientações básicas, até as mais recentes e refinadas técnicas de acessibilidade para interfaces altamente sofisticadas e interativas, incluindo referências para diretrizes e ferramentas orientadas a públicos com características específicas.

Se você é desenvolvedor ou designer de sites, sistemas ou aplicativos web, seja para interfaces de desktop ou dispositivos móveis, você encontrará vários capítulos direcionados ao seu trabalho. Eles possuem uma forma de apresentação, a partir das barreiras de acessibilidade, que o conduzirá às diretrizes técnicas por um caminho mais simples e amigável. E se já estiver familiarizado com as diretrizes, encontrará orientações práticas preciosas, tanto baseadas na literatura técnica, quanto na vasta experiência do autor.

Se você é provedor de conteúdo, ou desenvolvedor de agentes do usuário, também encontrará orientações relevantes e referências para diretrizes, bem como uma análise das ferramentas específicas, que possibilitarão que a acessibilidade das páginas web se mantenha íntegra até o usuário final.

Se você é gerente ou gestor de projetos web, de recursos humanos, políticas de diversidade, relacionamento com o cliente ou áreas afins, encontrará aqui conceitos, abrangência, público-alvo e informações gerenciais sobre implementação, manutenção, avaliação e divulgação da acessibilidade conquistada.

Se você trabalha com questões jurídicas, ou se é um usuário de acessibilidade com necessidade de defender os seus direitos, também encontrará informações sobre as leis vigentes e os procedimentos legais.

Quero, enfim, expressar minha gratidão por ter a honra e a satisfação de apresentar esta obra primorosa, que certamente se tornará, em pouco tempo, uma referência em acessibilidade na web, quer seja no âmbito acadêmico, como no âmbito técnico, além de contribuir para a divulgação em geral.

Boa leitura para vocês e muita acessibilidade para todos nós!

Lêda Lucia Spelta

Janeiro de 2020.

[1] Lêda Lucia Spelta é psicóloga formada pela UFRJ, com experiência em clínica e reabilitação. Tem formação em Terapia CranioSacral, Somatoemotional Release e Terapia Regressiva Integral. Foi uma das primeiras pessoas cegas a trabalhar com informática no Rio de Janeiro. Trabalha com acessibilidade desde 2001, como palestrante, instrutora e autora de artigos. Participou da Comissão Brasileira do Braille, coordenando a elaboração da Grafia Braille para a Informática, unificada para os países de língua portuguesa. É membro do Grupo de Trabalho em Acessibilidade do Escritório Brasil do W3C. Foi vencedora do Prêmio Nacional de Acessibilidade Todos@Web 2013.

SOBRE O AUTOR

REINALDO FERRAZ

Trabalho com desenvolvimento Web desde 1998. Comecei fazendo sites em HTML3, passando pelo Flash até entrar de cabeça no mundo dos padrões Web, no qual continuo envolvido até os dias atuais.

Comecei a me envolver com acessibilidade na Web em 2004, mesmo ano em que iniciei as atividades no NIC.br, devido à necessidade de compreender o Decreto 5296 de 2 de dezembro de 2004, que abordava a questão de sites acessíveis pela primeira vez. Participei de diversos workshops e cursos com especialistas, até me envolver com projetos relacionados à acessibilidade na Web nos sites do NIC.br.

Em 2011 fui convidado a fazer parte do time do Escritório Brasileiro do W3C (World Wide Web Consortium), que iniciou suas operações em 2008. Fui responsável pelos projetos relacionados à acessibilidade na Web, dos quais destacam-se a tradução autorizada das Diretrizes de Acessibilidade para Conteúdo Web, o Prêmio Nacional de Acessibilidade na Web, a coordenação do Grupo de Especialistas de Acessibilidade na Web do Ceweb.br e a Cartilha de Acessibilidade na Web do W3C Brasil.

Publiquei diversos artigos, tanto para revistas técnicas como acadêmicas, além de ter ministrado mais de cem palestras pelo Brasil e pelo mundo. Sou representante brasileiro das plenárias técnicas de acessibilidade na Web, publicações digitais e Web das

Coisas no W3C internacional.

Atualmente sou gerente de projetos no W3C Brasil e no Ceweb.br e continuo envolvido e apaixonado pela acessibilidade na Web. Este é o meu terceiro livro e o segundo sobre acessibilidade na Web.

Tenho muito material publicado espalhado pela internet. Os links abaixo agrupam parte deles.

- http://www.reinaldoferraz.com.br/
- https://w3c.br/
- https://www.ceweb.br/
- https://twitter.com/reinaldoferraz/
- https://www.instagram.com/reinaldoferraz/
- https://pt.wikipedia.org/wiki/Reinaldo_Ferraz
- https://www.linkedin.com/in/reinaldoferraz/

PARA QUEM ESTE LIVRO É RECOMENDADO

Este é um livro que eu gostaria de ter lido. Digo isso porque quando comecei a estudar sobre acessibilidade na Web eu tinha à minha disposição a documentação do W3C e muito material espalhado na rede.

O material da rede foi muito útil, mas debruçar-se sobre a documentação do W3C de ponta a ponta é um trabalho árduo de leitura e compreensão. Por isso quando escrevi este livro pensei em um guia para quem quer tirar dúvidas específicas (sejam elas de exemplos de código ou documentação) e em um livro de fácil leitura para entender sobre a acessibilidade digital e tirar da cabeça das pessoas que tornar um site acessível é complicado.

Tentei escrever este livro da forma mais simples, direta e didática para facilitar a compreensão de pessoas sem perfil técnico, mas ter um conhecimento básico relacionado a desenvolvimento Web, em especial HTML, CSS e JavaScript, torna mais fácil o consumo e a aprendizagem das boas práticas apresentadas.

Por isso, recomendo este livro para:

- **Desenvolvedores**: tanto para os que trabalham com aplicações Web em dispositivos móveis quanto desktop. Este livro atende desde o especialista que quer tirar uma dúvida até ao iniciante que precisa aprender por onde começar na acessibilidade.
- **Gestores**: gerentes de projeto e produto podem entender

de forma mais simples as barreiras de acesso apresentadas neste livro.

- **Estudantes**: principalmente para os de desenvolvimento de sistemas Web, para colocar as boas práticas de acessibilidade na rotina do desenvolvimento.
- **Professores**: pela possibilidade de apresentar aos alunos boas práticas no desenvolvimento de sistemas Web.
- **Interessados no tema**: o livro está escrito de uma forma de fácil entendimento, mesmo para aqueles sem conhecimento técnico.

Espero que este livro possa ajudar todos aqueles que querem fazer um projeto Web mais acessível.

AGRADECIMENTOS

Eu gostaria de agradecer a todos que me incentivaram neste projeto, que acreditam que eu ainda tenho muito a contribuir para uma Web mais acessível, mas quero mencionar algumas pessoas nesse trabalho:

Primeiramente agradeço à minha família, em especial à minha esposa Paula e ao meu filho Júlio, por sempre me estimularem, me apoiarem e pela paciência nas noites escrevendo e estudando para este livro. Agradeço também aos meus pais Cecília e Luiz pela educação e exemplos para me tornar o ser humano que sou hoje.

Aos meus mentores da acessibilidade digital, que acenderam a fagulha no meu coração: Lêda Spelta, Horácio Soares e o saudoso MAQ.

A todos os colegas do Ceweb.br, em especial ao Vagner Diniz, que sempre instiga minha curiosidade com projetos desafiadores para tornar a Web um ambiente muito mais humano.

Aos participantes do Grupo de Especialistas em Acessibilidade Digital (GT Acessibilidade) com quem aprendo todos os dias.

À amiga e grande especialista em acessibilidade digital Talita Pagani que me ajudou muito na revisão final desse trabalho.

Ao pessoal da Casa do Código, por toda a ajuda e paciência (Obrigado mesmo, Vivian!).

E para todos que se interessam pela acessibilidade na Web. Saibam que estou à disposição de vocês!

Sumário

1 Introdução à acessibilidade na Web **1**

1.1 Uma breve história da acessibilidade digital 3

1.2 A primeira foto na Web e a evolução da acessibilidade 7

1.3 Quem são os beneficiados pela Web acessível 12

2 Deficiência e tecnologia assistiva **19**

2.1 Deficiência visual - cegueira 20

2.2 Deficiência visual - baixa visão 23

2.3 Deficiência auditiva 25

2.4 Deficiência motora 26

2.5 Deficiência cognitiva/ neurológica 28

2.6 Como garantir o suporte a essas situações? 29

3 Diretrizes de acessibilidade na Web **31**

3.1 WCAG 33

3.2 WAI-ARIA 45

3.3 ATAG 59

3.4 UAAG 60

3.5 WCAG-EM 61

3.6 eMag 61
3.7 Section 508 63
3.8 ADA 64
3.9 Outras diretrizes técnicas importantes 64

4 Eliminando as principais barreiras de acesso 67
4.1 Estrutura 68
4.2 Navegação e interatividade 69
4.3 Design 69
4.4 Multimídia e conteúdo não textual 70

5 Eliminando barreiras - Estrutura 71
5.1 Título da página 72
5.2 Estrutura semântica 75
5.3 Tabela acessível 79
5.4 Idioma da página 81
5.5 Estrutura de cabeçalhos 82
5.6 Localização 83
5.7 Sequência de leitura com significado 84

6 Eliminando barreiras - Navegação e interatividade 87
6.1 Navegação por teclado 88
6.2 Acionamento por gestos 91
6.3 Atalhos de teclado e acionamento por voz 93
6.4 Saltar conteúdo repetido 94
6.5 Ordem de foco 96
6.6 Formulários 97
6.7 Acionamento e cancelamento de botão 111

6.8 Propósito do link 112
6.9 Tempo suficiente 114
6.10 Timeouts 115
6.11 Atualização automática de áreas do site 116
6.12 Ativação por movimento 118

7 Eliminando barreiras - Design 119
7.1 Design responsivo e acessibilidade 119
7.2 Orientação 123
7.3 Uso de cores 124
7.4 Redimensionar texto 129
7.5 Reorganização de conteúdo 131
7.6 Animações de interatividade 133
7.7 Conteúdo que cause convulsões 133
7.8 Escondendo e exibindo conteúdo 134

8 Eliminando barreiras - Multimídia e conteúdo não textual 136
8.1 Descrevendo imagens 136
8.2 Descrevendo imagens em SVG 144
8.3 Acessibilidade em áudio 148
8.4 Acessibilidade em vídeo 150
8.5 Controles de áudio e vídeo 154

9 Acessibilidade em CMS, Frameworks e outras aplicações para desenvolvimento Web 155
9.1 Wordpress 156
9.2 Joomla 157
9.3 Plone 158
9.4 React 158

9.5 Vue.js 160
9.6 Angular 162
9.7 Ember.js 163
9.8 JQuery 164
9.9 O que mais preciso saber? 164

10 Colocando acessibilidade na rotina do desenvolvimento 166
10.1 Planejamento 167
10.2 Desenvolvimento 168
10.3 Publicação de conteúdo 173
10.4 Próximos passos 176

11 Verificação de acessibilidade 177
11.1 Verificação automática de acessibilidade 178
11.2 HTML_Codesniffer 185
11.3 AChecker 190
11.4 Access Monitor Plus 201
11.5 Verificação manual de acessibilidade 205
11.6 Meu site está de acordo com as boas práticas. E agora? 208

12 Conformidade e questões legais 209
12.1 Declaração de conformidade 209
12.2 Fazendo sua própria declaração de conformidade 215
12.3 Selo de acessibilidade 215
12.4 Encontrei problemas em um site. O que faço? 216
12.5 Questões legais 217
12.6 Declaração não garante a acessibilidade completa 219
12.7 Ministério Público 219

12.8 Muito além de questões legais 221

13 A Web que queremos - de todos e para todos 222

14 Considerações finais 226

15 Referências 229

15.1 Padrões e documentos do W3C 229

15.2 Artigos e boas práticas 233

15.3 Leis e regulamentações 235

15.4 Tecnologia assistiva e de apoio 236

15.5 Ferramentas 239

15.6 Documentação de frameworks 241

15.7 Recursos diversos 244

Capítulo 1

INTRODUÇÃO À ACESSIBILIDADE NA WEB

Qual a primeira coisa que vem a sua cabeça quando você lê o título deste livro? Se a sua resposta for "pessoas com deficiência na Web" eu diria que você está parcialmente certo. Acessibilidade na Web está relacionada diretamente com a eliminação de barreiras em páginas para que pessoas com deficiência tenham autonomia na rede. Segundo o W3C, o consórcio que desenvolve os padrões para a Web, a definição de uma Web acessível é a seguinte:

"Acessibilidade na Web significa que sites, ferramentas e tecnologias são projetados e desenvolvidos para que pessoas com deficiências possam usá-los. Mais especificamente, as pessoas podem: a) perceber, entender, navegar e interagir com a Web; b) contribuir para a web." (https://www.w3.org/WAI/fundamentals/accessibility-intro/)

É importante contextualizar o que significa uma pessoa com deficiência. Segundo o Decreto nº 6949 de 25 de agosto de 2009, que promulga a Convenção Internacional sobre os Direitos das Pessoas com Deficiência, sua definição é a seguinte:

"Pessoas com deficiência são aquelas que têm impedimentos de

longo prazo de natureza física, mental, intelectual ou sensorial, os quais, em interação com diversas barreiras, podem obstruir sua participação plena e efetiva na sociedade em igualdades de condições com as demais pessoas." (http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/decreto/d6949.htm)

Esse texto também é utilizado na definição na Lei Brasileira de Inclusão (Lei nº 13.146, de 6 de julho de 2015). Essa solução se baseia no conceito de que as deficiências, sendo permanentes, necessitam de soluções permanentes para garantir a autonomia de todas as pessoas. Sendo assim, políticas públicas e ações vêm sendo tomadas ao longo dos anos para garantir cada vez mais a inclusão, eliminando barreiras que possam impedir sua participação na sociedade. Essas barreiras vão desde a falta de rampas para cadeirantes em edificações até a falta de acessibilidade em sistemas utilizados por pessoas com deficiência.

Nesta publicação, você vai entender por que aquela afirmação do primeiro parágrafo está parcialmente correta: acessibilidade na Web possibilita que pessoas com deficiência não tenham barreiras ao navegar em uma página ou utilizar uma aplicação na Web. Mas tornar o conteúdo acessível beneficia também outros grupos de usuários, que podem não ter deficiência mas utilizam a aplicação em diferentes cenários. Por exemplo, garantir um bom contraste entre texto e o fundo permite que pessoas com baixa visão consigam ler um texto com mais facilidade, mas permite também que pessoas que utilizam o celular na rua, sob incidência de sol na tela, tenham menos dificuldade em ler as informações na tela do celular.

Poderíamos enumerar vários exemplos e situações nas quais a acessibilidade na Web beneficia pessoas com deficiência e pessoas sem deficiência. A Iniciativa de Acessibilidade na Web do W3C (WAI) publicou uma série de vídeos que ilustram essa situação (https://www.w3.org/WAI/perspective-videos/).

Quero convidar você nesta breve introdução a compreender a importância da acessibilidade digital, para que você possa contribuir para um mundo mais inclusivo, para a construção de uma Web que seja de todos e para todos.

1.1 UMA BREVE HISTÓRIA DA ACESSIBILIDADE DIGITAL

A Web ainda é uma tecnologia jovem. Ela foi criada em 1989 por Tim Berners-Lee, no CERN[1]. Foi uma enorme revolução na forma como consumimos conteúdo digital, já que o conceito de hiperlinks (já criado anteriormente por Ted Nelson) da Web proporcionou a navegação entre documentos de forma muito mais simples. A acessibilidade na Web surgiu pouco tempo depois, mas antes vale a pena resgatar um pouco da história da acessibilidade digital.

No início, a Web era basicamente um conjunto de textos e links amontoados. Tim Berners-Lee desenvolveu não só a primeira página, mas também o primeiro navegador para acessá-la. Browsers que ajudaram a popularizar a Web, como o Mozaic, surgiriam somente em meados de 1993. A imagem a seguir mostra como era a navegação da primeira página no primeiro navegador (ainda é possível navegar na primeira página Web no link http://info.cern.ch/)

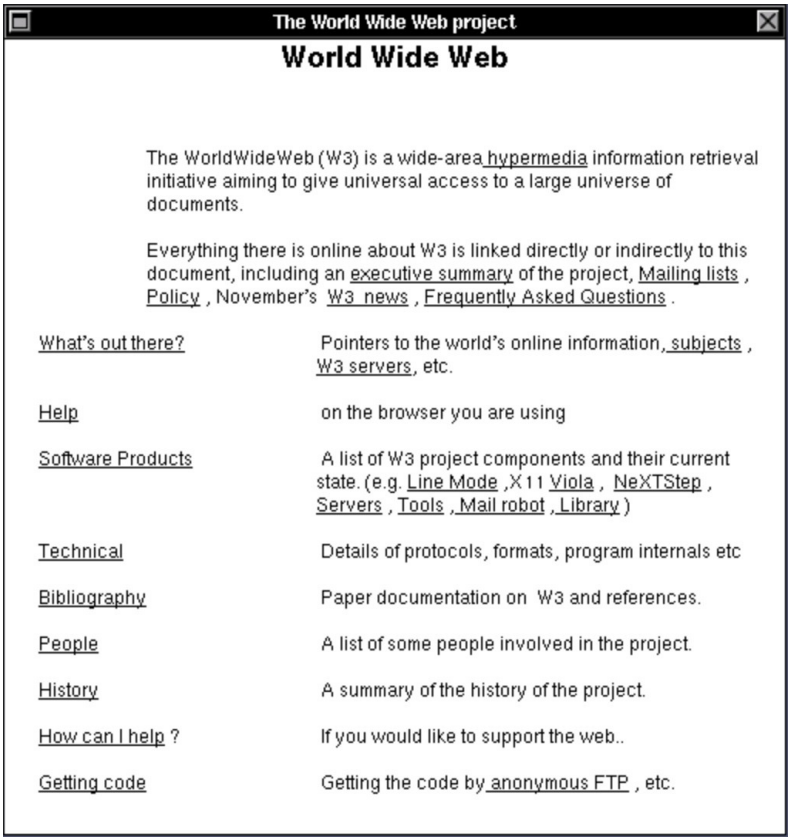

The World Wide Web project
World Wide Web
The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.
Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .
What's out there? Pointers to the world's online information, subjects , W3 servers, etc.
Help on the browser you are using
Software Products A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical Details of protocols, formats, program internals etc
Bibliography Paper documentation on W3 and references.
People A list of some people involved in the project.
History A summary of the history of the project.
How can I help ? If you would like to support the web..
Getting code Getting the code by anonymous FTP , etc.

Figura 1.1: Tela de navegador em modo texto exibindo a primeira página Web desenvolvida

Um dos primeiros leitores de tela para computadores pessoais surgiu por volta de 1986, alguns anos antes da criação da Web. O software chamado IBM Screen Reader foi criado por Jim Thatcher para uma interface para DOS e logo depois o próprio Jim liderou o desenvolvimento de do IBM Screen Reader/2, voltado para PCs com sistemas operacionais de interface gráfica (https://www.afb.org/afbpress/pubnew.asp?DocID=aw050207).

Um software leitor de tela, do inglês *screen reader*, é um sintetizador que transforma a informação selecionada na tela em áudio a ser transmitido para o usuário. Esse recurso, que é instalado no dispositivo do usuário (e não na página ou aplicação Web), é chamado de tecnologia assistiva (abordaremos mais detalhadamente tecnologia assistiva em um capítulo adiante).

Esse tipo de tecnologia assistiva possibilitou que pessoas com baixa visão ou cegas pudessem acessar o computador, mas para isso as informações deveriam estar disponíveis de forma que o leitor de telas pudesse compreender.

A Web começou a ganhar popularidade muito rapidamente graças ao trabalho dentro do W3C (instituição criada por Tim Berners-Lee para desenvolver e evoluir a Web), e a acessibilidade digital já começava a rondar essa nova forma de interação digital nos anos noventa. Foi no ano de 1997 que o W3C inaugurou sua iniciativa para acessibilidade na Web, a WAI, que viria a trabalhar diretamente para a garantia da acessibilidade nos padrões do W3C. Nenhum padrão sai como uma *W3C Recommendation* sem o crivo da Iniciativa de Acessibilidade.

No ano da sua inauguração, Tim Berners-Lee disse a frase que soa como um mantra da acessibilidade digital:

> *"O poder da Web está em sua universalidade. O acesso de todos, independentemente da deficiência, é um aspecto essencial."* (https://www.w3.org/Press/IPO-announce)

A partir daí, a preocupação com acessibilidade na Web começou a se tornar mais intensa. Surgiram as primeiras diretrizes de acessibilidade em 1998 e outras no decorrer dessas duas décadas, sendo elas tanto para agentes de usuário (como browsers e tecnologia assistiva) quanto para o desenvolvimento de ferramentas de autoria.

Uma série de documentos que orientam sobre como tornar a Web acessível surgiu depois dessa primeira versão das diretrizes de acessibilidade: WCAG 2.0 (2008) e 2.1 (2017), WAI-ARIA 1.0 (2014) e 1.1 (2017), ATAG 1.0 (2000) e 2.0 (2015) e outras siglas que veremos ao longo do livro.

Pudemos acompanhar também a popularização de recursos que possibilitam maior conforto, facilidade de leitura e compreensão diretamente no browser, com extensões e plugins que trazem desde a alteração de contraste na página e espaçamento de linhas até leitores de tela e sistemas de reconhecimento de voz dentro do próprio navegador (e não instalado no computador).

A indústria de hardware também tem prestado atenção a esse público. Óculos com câmeras que identificam objetos e pessoas e transmitem a informação em forma de áudio e sensores infravermelhos para identificar obstáculos nas ruas já não são mais exclusividade de ficção científica.

Mas o fator mais importante é o engajamento das pessoas em eliminar barreiras de acesso para pessoas com deficiência. Uma delas é a hashtag `#PraCegoVer` utilizada nas redes sociais para descrever as fotos para pessoas que não enxergam. Acessibilidade deixou de ser um tema exclusivo do universo das pessoas com deficiência e começou a se tornar um tema recorrente no dia a dia

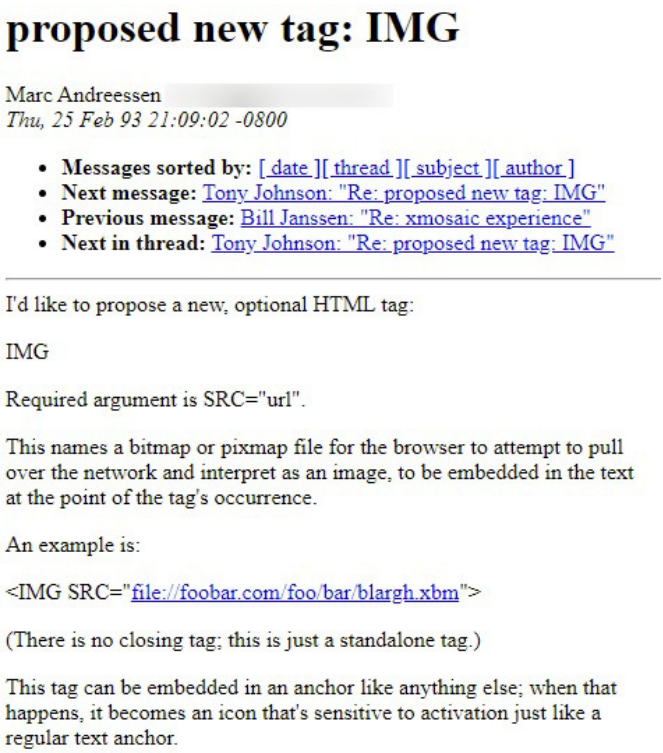
de um número maior de pessoas.

O fato é que hoje temos tecnologia e padrões suficientes para que não tenhamos mais desculpas para não tornar uma aplicação na Web acessível para todas as pessoas.

1.2 A PRIMEIRA FOTO NA WEB E A EVOLUÇÃO DA ACESSIBILIDADE

Conforme a Web evoluía, Tim Berners-Lee queria deixar sua invenção mais ampla e robusta. Em um experimento no CERN, ele pediu uma foto para um grupo de colegas que tinham uma banda (chamada Les Horribles Cernettes) para fazer alguns testes em sua invenção. A foto, publicada originalmente em 1992, é considerada a primeira foto na Web (que pode ser acessada pelo link https://en.wikipedia.org/wiki/File:Les_Horribles_Cernettes_in_1992.jpg). Ela não está disponível no livro pois não está publicada com licença de uso livre.

No ano seguinte, o grupo de discussão sobre a World Wide Web começava a discussão sobre o uso de elementos externos nas páginas. Em fevereiro de 1993, Marc Andreessen, o mesmo que inventou o navegador Mozaic, submetia uma proposta para adicionar um elemento de imagem na Web. A imagem a seguir mostra um trecho deste e-mail, mas ele está disponível no histórico da lista de discussão do W3C, em http://1997.webhistory.org/www.lists/www-talk.1993q1/0182.html)



proposed new tag: IMG

Marc Andreessen
Thu, 25 Feb 93 21:09:02 -0800

- **Messages sorted by:** [date][thread][subject][author]
- **Next message:** Tony Johnson: "Re: proposed new tag: IMG"
- **Previous message:** Bill Janssen: "Re: xmosaic experience"
- **Next in thread:** Tony Johnson: "Re: proposed new tag: IMG"

I'd like to propose a new, optional HTML tag:

IMG

Required argument is SRC="url".

This names a bitmap or pixmap file for the browser to attempt to pull over the network and interpret as an image, to be embedded in the text at the point of the tag's occurrence.

An example is:

<IMG SRC="file://foobar.com/foo/bar/blargh.xbm">

(There is no closing tag; this is just a standalone tag.)

This tag can be embedded in an anchor like anything else; when that happens, it becomes an icon that's sensitive to activation just like a regular text anchor.

Figura 1.2: Captura de tela do e-mail de Marc Andreessen sobre a sugestão de um elemento de imagem na Web

Tradução livre da mensagem:

Gostaria de propor uma nova tag HTML opcional:
IMG

O argumento necessário é SRC = "url".
Isso nomeia um arquivo bitmap ou pixmap para o navegador tentar puxar através da rede e interpretar como uma imagem, a ser incorporada no texto no ponto da ocorrência da tag.

Um exemplo é: `<IMG SRC = "arquivo://foobar.com/foo/bar/blargh.xbm">`

(Não há tag de fechamento; é apenas uma tag autônoma.)
Essa tag pode ser incorporada em uma âncora (link) como qualquer outra coisa; quando isso acontece, ela se torna um ícone sensível à ativação como um link de texto regular.

Dois anos depois, na mesma lista de discussão, Daniel N. Wood propunha a inserção de um atributo para texto alternativo das imagens. Esse texto serviria para descrever a imagem caso ela não fosse carregada. Seria uma das primeiras formas de garantir que um conteúdo não textual tivesse uma alternativa disponível. A próxima imagem mostra um trecho deste e-mail, mas ele também está disponível na íntegra na lista de discussão do W3C em http://1997.webhistory.org/www.lists/www-html.1995q2/0128.html.

A suggestion for alt text.

Daniel N. Wood
Sat, 15 Apr 1995 18:10:36 +0500

- **Messages sorted by:** [date][thread][subject][author]
- **Next message:** ADMINISTRATOR ROUTER _ "ERROR REPLY:"
- **Previous message:** Martian: "Re: Word wrapping"

I would like to suggestion an addition to the FIG tag. (And perhaps
the IMG tag.)

It seems to me that there are two common scenarios for alt text.

1) The alt text is a perfect substitute for the image. For example:

Replace IBM's logo with the word IBM.
Alt text for some sort of fancy graphical title or headline.
A set of hyperlinks which duplicate the function of an image
map.

2) The alt text describes the image. For example:

A photo of me.
An image map that is a _real_ map.
Charts or figures.

Wouldn't it be valuable to distinguish between these two cases?

Figura 1.3: Captura de tela do e-mail de Daniel N. Wood propondo o uso de um atributo de texto alternativo em imagens

Tradução livre da mensagem:

Eu gostaria de sugerir uma adição à tag FIG. (E talvez à tag IMG.)

Parece que há dois cenários comuns para o texto alternativo.

1) O texto alternativo é um substituto perfeito para a imagem. Por exemplo:

Substitua o logotipo da IBM pela palavra IBM.
Texto alternativo para algum título ou cabeçalho gráfico sofisticado.
Um conjunto de hiperlinks que duplicam a função de um mapa de imagem.

2) O texto alternativo descreve a imagem. Por exemplo:*

Uma foto minha.
Um mapa de imagem que é um mapa "real".
Gráficos ou figuras.

Não seria valioso distinguir entre esses dois casos?

A Web deixava de ser somente textos e hiperlinks. Com a chegada de imagens na Web, era realmente necessário o desenvolvimento de formas alternativas para acessar e consumir o conteúdo. E o grupo estava disposto a considerar isso uma preocupação importante.

1.3 QUEM SÃO OS BENEFICIADOS PELA WEB ACESSÍVEL

Consideraremos o termo “barreira de acesso” para ilustrar situações que podem impedir uma pessoa de acessar uma determinada informação em uma página Web ou interagir com ela. São os mais diversos cenários e pelos quais qualquer pessoa está sujeita a passar. Podemos listar os tipos de deficiência da seguinte forma:

- Deficiência visual
- Deficiência auditiva
- Deficiência motora
- Deficiência cognitiva ou neurológica

A partir de agora vamos compreender um pouco os principais aspectos referentes a cada uma delas.

Deficiência visual - cegueira

Quando pensamos na Web consideramos a orientação visual como a principal forma de leitura e da localização da informação em uma página. Porém, para as pessoas cegas, o acesso ao conteúdo na Web é feito por tecnologia assistiva que transforma toda a informação da página em som para o usuário. Pessoas cegas que usam leitores de tela precisam de informações semânticas em páginas que sejam transmitidas de forma adequada em som, por exemplo, uma imagem em uma página precisa de descrição, formulários precisam estar rotulados adequadamente e links devem ser claros para que o usuário consiga compreender e interagir com a página.

Deficiência visual - baixa visão

Baixa visão consiste na diminuição da acuidade visual a ponto de uma perda considerável da visão. A maioria das pessoas com baixa visão precisa aumentar o tamanho das fontes ou utilizar lupas para ler o conteúdo em uma página Web.

Deficiência visual - daltonismo

Apesar de não haver consenso se daltonismo é ou não uma deficiência, pessoas que não conseguem enxergar cores podem ter dificuldades para compreender informações somente em cor, por exemplo, uma tabela de ônibus que marca os atrasados em vermelho e os pontuais em verde.

Deficiência auditiva - surdez

Imagine uma pessoa que não consegue escutar acessando a Web cheia de YouTubers e vídeos em redes sociais. Pessoas surdas podem ter dificuldades em compreender o conteúdo de uma obra ou aplicação em audiovisual. Ela necessita que o conteúdo tenha legendas, transcrição ou tradução para LIBRAS (Língua Brasileira de Sinais) para não ter barreiras ao consumir esse tipo de conteúdo na rede.

Deficiência auditiva - baixa audição

Pessoas com baixa audição têm redução da sua capacidade de ouvir e podem necessitar dos mesmos recursos que uma pessoa surda utiliza para consumir conteúdo digital.

Deficiência motora

Essa deficiência interfere diretamente na interação com o conteúdo na Web. Pessoas com deficiência motora, especialmente aqueles que não se movimentam do pescoço para baixo, podem não conseguir navegar em páginas complexas, que dependam de mouse ou de habilidades específicas com o uso de computadores.

Deficiência cognitiva ou neurológica

Esse tipo de deficiência envolve um amplo espectro relacionados ao processo cognitivo. Está relacionado a funções cerebrais que envolvem desde transtornos cognitivos até doenças mentais. Abrange condições diversas que afetam o sistema nervoso e impactam na forma como as pessoas escutam, se movimentam, enxergam, falam e compreendem as informações. Porém, não necessariamente afetam o desenvolvimento intelectual (Fonte: https://www.w3.org/WAI/people-use-web/abilities-barriers/).
Exemplo de condições dentro deste espectro são: Transtorno de Déficit de Atenção e Hiperatividade (TDAH), Transtorno do Espectro do Autismo (TEA), Dislexia, Síndrome de Down, epilepsia, depressão, transtorno bipolar, problemas de memória de curto e longo prazo, entre outros. Assim como o daltonismo, não há um consenso sobre o enquadramento de várias destas condições como deficiência, embora transtornos como TEA e Síndrome de Down estejam amparados pelas LBI. Essas pessoas podem ter dificuldades em navegar por páginas que piscam, páginas cheias de anúncios, não conseguir ler textos longos ou com espaçamento irregular ou até não preencher um formulário em páginas com muitas distrações.

A acessibilidade na Web beneficia esse grande número de pessoas, que somam quase 25% da população brasileira segundo o último Censo do IBGE de 2010 (https://censo2010.ibge.gov.br/noticias-censo?id=3&idnoticia=2170&view=noticia). Esse número representa mais de 45 milhões de pessoas no Brasil (considerando o número atual de brasileiros, esses quase 25% da população estariam em torno de 50 milhões de pessoas). Em 2020, o próximo Censo deve trazer números atualizados. No mundo, segundo a ONU, são mais de 1 bilhão de pessoas com algum tipo de deficiência.

Quando esse número é colocado em conjunto com o número de pessoas que usam a internet, a coisa fica mais complicada. principalmente porque nem todos os países possuem dados sobre o número de pessoas com deficiência utilizando a internet. Segundo o *Office for National Statistics* do Reino Unido em uma pesquisa feita em maio de 2015 (https://www.theguardian.com/technology/2015/jun/29/disabled-people-internet-extra-costs-commission-scope), 27% dos adultos com deficiência no Reino Unido nunca tinham usado a internet. Comparando esse número com os 11% de adultos sem deficiência que nunca usaram a internet, esse percentual impressiona.

Dentro do número de pessoas com deficiência no Brasil e no mundo está uma parte da população que muitas vezes não se considera pessoa com deficiência, mas que pode ter alguma barreira ao acessar conteúdo na rede. São elas:

Pessoas idosas

Conforme a idade vai avançando a nossa capacidade e

habilidade para o uso do mouse pode diminuir, bem como nossa acuidade visual, auditiva e cognitiva.

O CETIC.br, ligado ao Comitê Gestor da Internet e ao NIC.br, publica estatísticas de uso da internet no Brasil desde 2015. Suas pesquisas mostram que o número de pessoas idosas que nunca usou a internet vem caindo: de 98% em 2005 para 78% em 2014. Esse número também cai quando verificamos o número de pessoas com 60 anos ou mais que não possuem telefone celular: em 2015 eram 77% e em 2014 esse número passou para 36%. Isso significa que mais pessoas, principalmente idosas, estão usando a internet e telefones celulares. Todos os dados da pesquisa podem ser consultados em http://data.cetic.br/.

Deficiência temporária

Pessoas que fizeram cirurgias nos olhos podem fazer uso de tecnologia assistiva para acessar informações em páginas Web e deixar de usá-las quando estiver recuperada, bem como uma pessoa com o braço quebrado que pode fazer o uso da outra mão com pouca habilidade para navegar.

Usuários de dispositivos móveis

Ler uma informação em uma tela pequena já é difícil. Agora imagine ler com incidência de sol na tela do seu telefone celular? Isso pode ser complicado e muitas vezes impeditivo caso não seja possível aumentar as letras do visor do seu telefone celular.

Ambientes silenciosos

Assistir um filme sem som e com legendas habilitadas pode ser

a única forma de assistir a um conteúdo audiovisual em um ambiente silencioso quando não se tem um fone de ouvido.

São diversos os fatores que nos trazem a importância da acessibilidade digital. Essas primeiras páginas deste livro serviram apenas para que você pudesse compreender de forma simples e direta que acessibilidade na Web não envolve apenas o outro. Não é algo somente para uma pessoa com deficiência. Ela serve para qualquer pessoa, seja ela um desconhecido, um parente ou mesmo para você.

Entendeu agora por que sua resposta no início deste capítulo estava parcialmente certa? Acessibilidade na Web significa garantir o acesso a todas as pessoas, independente de alguma limitação, seja ela permanente ou temporária. Na verdade, o conceito de acessibilidade na Web mais adequado a essa publicação é um desenvolvido pelo W3C Brasil em sua Cartilha de Acessibilidade na Web:

"Acessibilidade na web é a possibilidade e a condição de alcance, percepção, entendimento e interação para a utilização, a participação e a contribuição, em igualdade de oportunidades, com segurança e autonomia, em sítios e serviços disponíveis na web, por qualquer indivíduo, independentemente de sua capacidade motora, visual, auditiva, intelectual, cultural ou social, a qualquer momento, em qualquer local e em qualquer ambiente físico ou computacional e a partir de qualquer dispositivo de acesso." (http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html)

Se não levantarmos a bandeira da acessibilidade digital hoje,

pode ser que no futuro tenhamos barreiras de acesso em páginas que nós mesmos construímos. E que depois podemos ter dificuldade em derrubá-las.

Por isso, convido você agora a conhecer mais de perto a acessibilidade na Web e aprender na prática como eliminar as principais barreiras de acesso de um site ou aplicação Web.

[1]CERN: Organização Europeia para a Pesquisa Nuclear, do françês *Conseil Européen pour la Recherche Nucléaire*, é o laboratório de física de partículas localizado em Genebra. É lá que está localizado o maior acelerador de partículas do mundo.

Capítulo 2

DEFICIÊNCIA E TECNOLOGIA ASSISTIVA

No primeiro capítulo, abordamos rapidamente algumas deficiências e como elas são impactadas com a acessibilidade digital. Neste capítulo, o objetivo é fazer uma relação mais ampla com deficiência, barreiras de acesso, tecnologia assistiva e como garantir que as pessoas consigam usar a Web com mais autonomia.

O espectro de deficiências que envolve barreiras de acesso em páginas e aplicações Web são enormes. Vão desde impossibilidade de enxergar até dificuldades de movimentação dos membros superiores. Em determinados casos, é impossível utilizar um dispositivo computacional (como tablet, celular ou computador) sem algum recurso de tecnologia assistiva.

Segundo a Lei Brasileira de Inclusão, Tecnologia Assistiva são "*produtos, equipamentos, dispositivos, recursos, metodologias, estratégias, práticas e serviços que objetivem promover a funcionalidade, relacionada à atividade e à participação da pessoa com deficiência ou com mobilidade reduzida, visando à sua autonomia, independência, qualidade de vida e inclusão social.*" (http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2015/Lei/L13146.htm).

Para uma melhor compreensão de tecnologia assistiva no âmbito da Web, podemos dizer que tecnologia assistiva são recursos de hardware e software que possibilitam o acesso de pessoas com deficiência ao dispositivo computacional e à Web.

Os recursos de tecnologia assistiva vão desde displays braille, que exibem o texto da tela em uma régua próxima ao teclado com pinos que representam o alfabeto braille, até software de ampliação de tela ou contraste. A seguir listamos certas deficiências e como cada uma delas lida com recursos de tecnologia assistiva (e como você pode usar esses recursos e testar em suas aplicações).

2.1 DEFICIÊNCIA VISUAL - CEGUEIRA

Pessoas que não conseguem enxergar a tela do computador, tablet os smartphone utilizam software que lê o código da página e transmite as informações em forma de áudio. São os *leitores de tela*. Apesar do poder tecnológico desse recurso, nem sempre os usuários conseguem navegar devido a problemas de codificação nas páginas. Por isso, a acessibilidade na Web é importante para garantir que as ferramentas compreendam o código e transmitam a informação correta ao usuário.

Em leitores de tela para computadores, o uso de atalhos de teclado é importante porque facilita a navegação. Na maioria dos leitores o uso da tecla "H" possibilita a navegação por cabeçalhos, por exemplo. Para que possam compreender cada um desses atalhos, segue uma lista de links de atalhos dos principais leitores de tela para computadores utilizados no Brasil:

- **JAWS** (Pago. Grátis para testar. Para Windows)

Download:
http://www.freedomscientific.com/products/software/jaws/

Lista de atalhos:

https://support.freedomscientific.com/Content/Documents/Manuals/JAWS/Keystrokes.pdf

- **NVDA** (Grátis. Para Windows)

Download: https://www.nvaccess.org/download/

Lista de atalhos:

https://www.nvaccess.org/files/nvda/documentation/userGuide.html?#BrowseMode/

- **ORCA** (Grátis. Para Linux)

Download: https://wiki.gnome.org/Projects/Orca/

Lista de atalhos:

https://help.gnome.org/users/orca/stable/commands.html.en/

- **Chromevox** (Grátis. Para Chromebook e plugin para o Google Chrome)

Download:
https://chrome.google.com/webstore/detail/chromevox/kgejglhpjiefppelpmljglcjbhoiplfn?hl=pt-BR

Lista de atalhos:

https://support.google.com/chromebook/answer/7031755?hl=pt-BR

Além dos leitores de tela para computadores/notebooks existem os para dispositivos móveis. Os ambientes iOS e Android dispõem de leitores de tela que também possuem documentação para compreender os toques e como configurar o software. São eles:

- **VoiceOver** (Nativo do IOs)

Download:
https://www.apple.com/br/accessibility/iphone/vision/

Lista de gestos: https://help.apple.com/iphone/10/?lang=pt-br#/iph75e97af9b

- **Talkback** (Grátis. Para Android)

Download: https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback&hl=pt

Lista de gestos:

https://support.google.com/accessibility/android/answer/6151827

Instalar alguns desses leitores de tela pode ajudar a identificar diversas barreiras de acesso em páginas que uma verificação automática pode não encontrar, por exemplo, se as imagens estão com a descrição adequada ou se existe algo que impeça a navegação por teclado.

Na maioria dos casos basta instalar o software e começar a usar. Lembre-se de que usuários de leitores de tela não costumam ter a referência visual, então a navegação deve ser feita preferencialmente por teclado (sem uso do mouse) nos

computadores.

Existem pesquisas que nos ajudam a compreender como as pessoas com deficiência utilizam tecnologia assistiva. A organização internacional *WebAim* publica periodicamente uma pesquisa com usuários de leitores de tela. Os números mostram as dificuldades encontradas, as principais barreiras de acesso e como elas navegam com leitores de tela. Essa pesquisa está disponível em https://webaim.org/projects/screenreadersurvey7/.

No Brasil a equipe de acessibilidade Everis Brasil promoveu um estudo similar, disponível em https://estudoinclusivo.com.br/.

2.2 DEFICIÊNCIA VISUAL - BAIXA VISÃO

Pessoas com baixa visão precisam de recursos que ampliem ou melhorem a forma de leitura em uma página Web. Podemos considerar como baixa visão um espectro muito grande de situações, como pessoas com miopia, glaucoma, retinopatia e até daltonismo.

Essas pessoas costumam ter dificuldades em enxergar letras pequenas, por isso o uso de recursos para aumentar a tela ou as fontes da página costuma ser habitual. Alguns sites utilizam esse recurso de aumento de fontes embutido enquanto outros apenas orientam ao usuário para utilizar “ctrl/option +” para aumentar as fontes no computador e fazer o movimento de pinça em dispositivos móveis.

Existem ferramentas que permitem que o usuário aumente toda a sua tela. São recursos que aumentam não só o conteúdo no navegador mas em tudo o que é exibido na tela. São conhecidas

como Lupas Digitais.

Algumas pessoas utilizam recursos de melhora de contraste para enxergar textos nas páginas Web. Esse também é um recurso utilizado em algumas páginas, mas o usuário tem à sua disposição alguns plugins de navegadores que permitem que ele altere o contraste em todos os sites.

No caso do daltonismo, quando a pessoa não enxerga alguns espectros de cores (vermelho, verde, azul ou todas as cores) existem recursos em navegadores para minimizar essa dificuldade, permitindo a alteração do contraste e das cores da página. Mas para que essa ferramenta funcione é necessário que exista um bom contraste entre o texto e o fundo (que será explicado melhor nos capítulos de eliminação de barreiras em sites e aplicações Web).

Algumas aplicações que servem ao propósito de eliminar barreiras de usuários de baixa visão:

- **LentePró** (Grátis. Para Windows)

Download: http://intervox.nce.ufrj.br/dosvox/dosvox.html

- **Magic** (Pago. Para Windows)

Download:
http://www.freedomscientific.com/products/software/magic/

- **High Contrast** (Grátis. Plugin para Google Chrome)

Download: https://chrome.google.com/webstore/detail/high-contrast/djcfdncoelnlbldjfhinnjlhdjlikmph?hl=pt-BR

- **Text Legibility** (Grátis. Plugin para Firefox)

Download: https://addons.mozilla.org/pt-BR/firefox/addon/text-legibility/?src=search

Alguns sistemas operacionais já oferecem ferramentas similares, como o aplicativo Lupa, nativo do sistema operacional Windows.

2.3 DEFICIÊNCIA AUDITIVA

Pessoas surdas ou com baixa audição podem ter dificuldades com conteúdo transmitido somente em áudio. Vídeos, podcasts ou programas de rádio podem limitar o acesso de pessoas que não conseguem ouvir ou compreender o áudio exibido. É importante ressaltar que existe uma grande parcela de pessoas surdas que tem dificuldade em compreender o português, e se comunica preferencialmente em LIBRAS (Língua Brasileira de Sinais).

Felizmente a evolução de recursos e tecnologia em favor da pessoa surda vem ganhando espaço, com tradutores de LIBRAS e recursos de transcrição para que pessoas que não escutam possam compreender áudio e texto exibido em uma página.

A tradução para LIBRAS segue os princípios de tradutores convencionais para outros idiomas, já que ela tem uma gramática própria diferente da língua portuguesa.

Tradutores de LIBRAS servem para que o usuário selecione um texto e literalmente traduza-o para a língua de sinais. Dessa forma os usuários que não conseguem escutar podem se comunicar em tempo real com ouvintes, já que alguns desses aplicativos fazem o reconhecimento de áudio.

Ferramentas de transcrição também têm se popularizado. Existem aplicativos que podem ser instalados no computador/smartphone ou até utilizado na Web. Essa ferramenta captura o áudio executado e transcreve o texto para o usuário.

Algumas dessas aplicações são as seguintes:

- **Handtalk**

(Tradutor de LIBRAS): https://www.handtalk.me/

- **VLibras**

(Tradutor de LIBRAS): http://www.vlibras.gov.br/

- **WebCaptioner**

(Ferramenta de transcrição): https://webcaptioner.com/

2.4 DEFICIÊNCIA MOTORA

As deficiências motoras que impactam principalmente os usuários de computadores, tablets e smartphones são aquelas que impedem o movimento dos braços, seja devido à tetraplegia, amputação ou outras situações nas quais o usuário não consegue utilizar os braços e as mãos para interagir com o dispositivo.

Nesses casos, os recursos de tecnologia assistiva vão mais na linha de auxiliar o usuário a utilizar o dispositivo, como ponteiras utilizadas na testa ou controladas com a boca (conhecidas como *mouthstick*), mouses e teclados adaptados e software de reconhecimento de voz ou movimentos (alguns utilizam movimentos dos olhos ou nariz para controlar o mouse, por

exemplo).

Para esses usuários o significado dos elementos em uma página ou aplicação é fundamental. Um sistema de controle por voz pode não conseguir acionar um botão em uma aplicação se aquele elemento não está marcado como tal. No caso de controles por movimento, o tamanho dos elementos interativos interfere na experiência do usuário, já que tocar em um botão pequeno com o mouse é muito mais fácil do que com uma aplicação que acompanha os movimentos da sua cabeça ou olhos.

Apesar de existir uma série de aplicações para facilitar a navegação de pessoas com mobilidade reduzida, em alguns casos o usuário não faz uso de nenhum recurso como esses, mas apenas de navegação pelo teclado ou toque. Fica a critério do usuário escolher a melhor ferramenta/tecnologia para auxiliá-lo na navegação.

Conheça algumas aplicações que beneficiam o acesso de pessoas com deficiência motora:

- **Configuração de navegação da Apple** (Para iPhone e iPad)

Possibilita que o usuário controle o dispositivo com movimentos simples de cabeça.

https://support.apple.com/en-us/HT201370

- **EVA Facial Mouse** (Para dispositivos Android)

Aplicativo que permite o controle de uma ponteira de mouse com movimentos da cabeça.

https://play.google.com/store/apps/details?

id=com.crea_si.eviacam.service

- **Siri** (Para iPhone e iPad)

Assistente pessoal da Apple controlado por comandos de voz.

https://www.apple.com/br/siri/

- **Pesquisa por voz do Google** (Para dispositivos Android)

Possibilita o controle do dispositivo por voz, através do "OK Google".

https://support.google.com/websearch/answer/2940021?co=GENIE.Platform%3DAndroid&hl=en&oco=0

2.5 DEFICIÊNCIA COGNITIVA/ NEUROLÓGICA

Para quem tem dificuldade em compreender e utilizar a Web devido a questões relacionadas a limitações cognitivas, o ideal é que essa pessoa consiga adaptar o ambiente conforme sua necessidade, seja ela eliminando alinhamentos de texto, aumentando entrelinhas ou colocando réguas para facilitar a leitura.

Interrupções no fluxo de leitura são muito comuns, principalmente em páginas com muitos elementos em movimento. Por isso, os usuários necessitam de recursos que deixam somente o texto visível ao usuário ou que tornem a leitura mais confortável.

Existem aplicações e plugins de navegadores que auxiliam essas pessoas, no sentido de tornar o uso da Web mais fácil e legível.

Alguns desses aplicativos relacionados a melhorar a experiência de pessoas com dislexia trazem uma série de recursos que tornam a leitura mais fácil para o usuário:

- **WebHelp** (Plugin para Google Chrome)

Disponibiliza réguas, marcadores e possibilita a customização para tornar a leitura mais confortável para pessoas com dislexia.

https://chrome.google.com/webstore/detail/webhelp/pjnhjelpkdoihfjeeemmahpdbmgliboo?hl=pt-BR

- **Dyslexia Formatter** (Plugin para Google Chrome)

Possibilita alterar recursos do navegador que facilitam a navegação por pessoas com dislexia .

https://chrome.google.com/webstore/detail/dyslexia-formatter/kggkghfhlppjclojgphbploiaipgogoc?hl=pt-BR

2.6 COMO GARANTIR O SUPORTE A ESSAS SITUAÇÕES?

Aqui talvez o fator de empatia seja o mais importante para considerar o acesso de todas as pessoas. A partir do momento em que sabemos que uma pessoa cega utiliza um leitor de telas para ler ou que uma pessoa com baixa visão utiliza recursos de aumentar e diminuir fontes, precisamos considerar os seguintes aspectos no desenvolvimento de uma aplicação:

- Instalar leitores de tela e fazer testes;
- Garantir que o aumento de fontes não impossibilite a leitura ou navegação;

- Verificar o contraste entre o texto e o fundo da página;
- Garantir que não existem barreiras para a navegação por teclado;
- Evitar bloquear a customização pelo usuário (por exemplo, evitar o bloqueio de zoom em dispositivos móveis).

Sobre as aplicações de controle de contraste, aumento de fontes e tradutores de LIBRAS embutidos no site, não existe uma regra permitindo ou proibindo o seu uso. Da mesma forma que existem recursos de tecnologia assistiva que permitem que o usuário utilize o seu próprio software em uma página bem codificada, alguns sites preferem dar ao usuário a opção de controlar esses recursos.

No decorrer do livro você vai compreender que a resposta a essa pergunta está na eliminação de barreiras de acesso, que serão detalhadas mais adiante.

Capítulo 3

DIRETRIZES DE ACESSIBILIDADE NA WEB

São os principais documentos que orientam o desenvolvimento da Web sem criar barreiras de acesso para pessoas com deficiência. Na maioria dos casos são documentos técnicos, mas que permitem uma identificação clara das barreiras de acesso para definir as medidas a serem tomadas.

É importante ressaltar que as diretrizes de acessibilidade fazem uso do material disponível dentro de cada documentação técnica de padrões na Web. Na documentação HTML5.2, por exemplo, ao citar o elemento `<img>`, são descritos todos os seus atributos e valores relacionados a ele, incluindo os que são necessários para a acessibilidade, como o `alt`, necessário para proporcionar texto alternativo para imagens.

Nesse sentido, as principais diretrizes apenas orientam o desenvolvedor a seguir corretamente a documentação técnica. É muito comum perceber em tutoriais, cursos ou até artigos sobre desenvolvimento Web que falam de HTML5 mas não falam dos recursos de acessibilidade que envolvem cada elemento. Em casos mais complexos, como aplicações ou widgets, talvez seja necessário considerar recursos adicionais na aplicação, mas que não passam

de atributos dentro do código HTML da página.

O capítulo destinado a apresentar as diretrizes de acessibilidade vai abordar não só a documentação técnica do W3C, mas também diretrizes utilizadas em outros países para garantir *compliance* com a acessibilidade na Web.

Para compreender um pouco melhor a relação das principais diretrizes de acessibilidade do W3C com o conteúdo, desenvolvedores e usuários, o W3C criou uma imagem que ilustra bem essa situação:

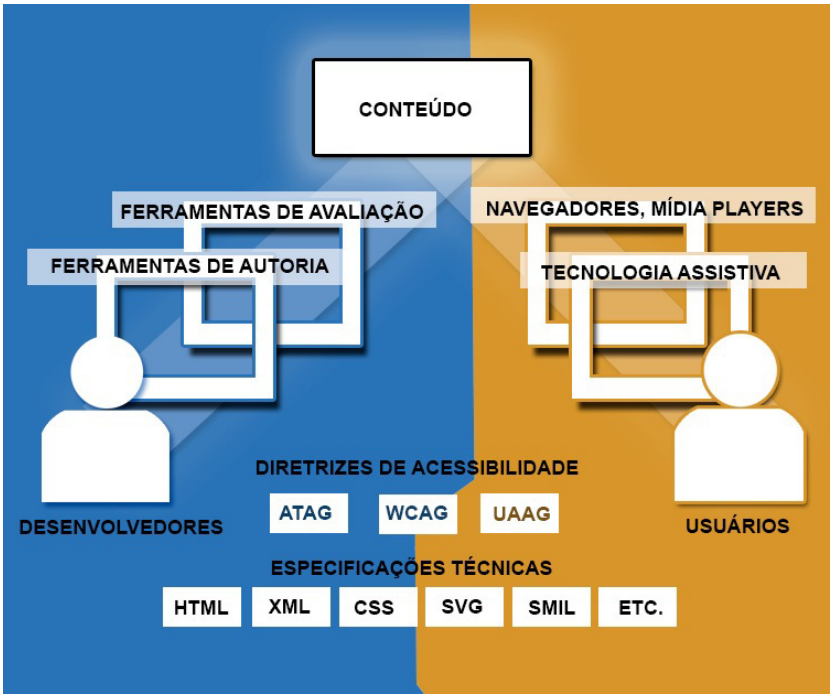


Figura 3.1: Imagem mostrando a relação do usuário e do desenvolvedor com diretrizes e conteúdo na Web

A imagem mostra que o usuário faz uso de navegadores e players de mídia, mas tem uma camada de tecnologia assistiva. Todos esses recursos fazem uso das UAAG. Já os desenvolvedores, que usam ferramentas de autoria para criação de conteúdo e ferramentas de verificação, fazem uso dos documentos ATAG e WCAG. As especificações técnicas se dividem, sendo HTML, XML e CSS do lado do desenvolvedor e SVG e SMIL do lado do usuário. Tudo isso para atingir de forma plena o conteúdo, que será produzido corretamente pelo desenvolvedor e consumido sem barreiras pelo usuário.

Imagem produzida com base no original do W3C, disponível em
https://www.w3.org/People/shadi/Talks/2007/0924/WAI/Images/specs.png

3.1 WCAG

O Web Content Accessibility Guidelines pode ser considerado o principal documento de acessibilidade na Web no qual o desenvolvedor se baseia para tornar o conteúdo acessível. São as Diretrizes de Acessibilidade para Conteúdo Web (https://www.w3.org/TR/WCAG/), que atualmente (abril de 2019) está em sua versão 2.1.

As WCAG orientam de forma muito simples como identificar e implementar técnicas que eliminam barreiras de acesso para pessoas com deficiência. O documento é extenso, o que pode causar desconforto para quem busca uma solução rápida para uma eventual correção de acessibilidade. Pensando nisso, este livro tem como objetivo ser mais direto nesse sentido, de facilitar a vida do

desenvolvedor quando necessitar de um guia rápido baseado nas diretrizes de acessibilidade do W3C. No capítulo 4, as técnicas e diretrizes deste documento serão explicadas de forma mais compreensível.

Para entender o documento (e seus documentos de apoio), é necessário conhecer suas camadas de orientação. Essas camadas distinguem conteúdo relacionado a desenvolvedores, gestores, legisladores e público em geral interessado em acessibilidade. São as seguintes:

- **Princípios:** são os quatro princípios que constroem a base da acessibilidade na Web. A partir deles, as camadas são separadas de forma mais ampla. Os princípios são os seguintes:
    - **Perceptível**: *As informações e os componentes da interface do usuário devem ser apresentados em formas que possam ser percebidas pelo usuário.* Isso significa que o conteúdo não pode ser invisível para mais de um sentido do usuário. Um bom exemplo para este princípio é considerar que qualquer conteúdo não textual precisa ter uma alternativa em texto, seja ele uma imagem, um campo de formulário etc.
    - **Operável**: *Os componentes de interface de usuário e a navegação devem ser operáveis.* Para tanto, o usuário não deve ter impedimentos para utilizar a interface, seja ele por um computador, smartphone ou tablet. Para exemplificar esse princípio, as diretrizes apontam a importância da navegação por teclado, já que nem todas as pessoas têm facilidade com o uso do mouse.

    - **Compreensível**: *A informação e a operação da interface de usuário devem ser compreensíveis.* Este princípio toca em aspectos que não são necessariamente técnicos, já que está relacionado à compreensão do usuário. Uso adequado de cores, abreviaturas e outras diretrizes relacionadas a conteúdo fazem parte deste princípio.
    - **Robusto**: *O conteúdo deve ser robusto o suficiente para poder ser interpretado de forma confiável por uma ampla variedade de agentes de usuário, incluindo tecnologias assistivas.* Basicamente, isso representa que a aplicação deve ser bem construída, de forma a ser acessível para uma gama maior de navegadores e tecnologia assistiva. O uso correto dos padrões Web já ajuda bastante a atingir este princípio.

- **Diretrizes**: são os itens que fornecem os objetivos básicos que devem ser atingidos dentro de cada princípio. Elas agrupam temas mais específicos, como as orientações sobre todos os componentes do site (imagens, vídeo, formulários etc.). A partir das diretrizes começa a ficar mais claro o caminho para tornar o conteúdo acessível. Dentre elas podemos listar e destacar algumas delas conforme estão documentadas no WCAG 2.1:

    - **Diretriz 1.1 Alternativas em texto:** *Fornecer alternativas textuais para qualquer conteúdo não textual, para que possa ser transformado em outras formas de acordo com as necessidades dos usuários, tais como impressão com tamanho de fontes maiores, braille, fala, símbolos ou linguagem mais simples.*

- **Diretriz 2.1 Acessível por teclado:** *Fazer com que toda funcionalidade fique disponível a partir de um teclado.*
- **Diretriz 3.1 Legível:** *Tornar o conteúdo do texto legível e compreensível.*
- **Diretriz 4.1 Compatível:** *Maximizar a compatibilidade entre os atuais e futuros agentes de usuário, incluindo tecnologias assistivas.*

- **Critérios de sucesso**: detalha como obter sucesso em cada diretriz, com descrições mais claras e diretas para determinar as soluções técnicas a serem utilizadas. Os critérios de sucesso possuem níveis de conformidade da seguinte forma:

  - **Nível A:** nível mínimo de conformidade. Atingindo este nível um grande conjunto de barreiras de acesso são eliminadas, possibilitando que pessoas com certos tipos de deficiência consigam acessar de forma plena.
  - **Nível AA:** com os critérios de sucesso de níveis A e AA, sua aplicação consegue eliminar mais barreiras de acesso, atingindo um público maior do que somente os contemplados pelo nível A.
  - **Nível AAA:** satisfaz os critérios dos níveis anteriores, possibilitando que mais pessoas sejam contempladas com a acessibilidade da aplicação, porém é o mais difícil de se atingir. Não se recomenda utilizar o nível AAA como política de acessibilidade para sites inteiros, mas em algumas áreas, quando necessário, pode ser bastante útil.

- **Técnicas suficientes e recomendadas**: aqui colocamos a mão na massa no código. Essas técnicas orientam como compreender os critérios de sucesso e como solucioná-los, com links e referências técnicas dentro do próprio documento do W3C.

Versões das WCAG

A documentação de acessibilidade na Web não é nova. Ela já existe desde 1999 com o mesmo objetivo: orientar como tornar as páginas Web acessíveis. Conforme o tempo foi passando e a Web evoluindo, essa documentação também sofreu alguns ajustes importantes para se manterem atualizados.

WCAG 1.0 (1999) - https://www.w3.org/TR/WAI-WEBCONTENT/

A primeira versão das diretrizes de acessibilidade trazia uma extensa lista de técnicas diretas para tornar o conteúdo acessível. Não possuía a abordagem por princípios, o que dificultava a compreensão de leitores com pouco conhecimento técnico.

Nessa época, a Web ainda engatinhava e nem imaginávamos que existiria áudio e vídeo nativo em navegadores. O máximo que esta documentação abordava em conteúdo multimídia era relacionado ao uso de *applets* na aplicação. Mesmo assim, o documento já abordava o conceito de não direcionar as diretrizes para um dispositivo específico, possibilitando que a evolução tecnológica fosse abraçada pela documentação conforme surgiam novos dispositivos com acesso à Web.

Apesar de ser uma documentação antiga seu conteúdo ainda é

muito útil, já que a base das diretrizes estão nele. As técnicas para tornar imagens, tabelas ou formulários acessíveis utilizadas neste documento ainda são válidas em grande maioria.

De qualquer forma, o W3C encoraja a utilização da versão mais recente das diretrizes.

WCAG 2.0 (2008) - https://www.w3.org/TR/WCAG20/

Foi a primeira grande atualização das diretrizes de acessibilidade. Trouxe os princípios que norteiam as diretrizes e tornam sua leitura mais "humana", deixando os termos técnicos dentro de cada documento de suporte.

Esta versão do documento tem uma grande importância para a acessibilidade na Web, pois traz a temática de conteúdo dinâmico e multimídia em sua especificação.

Além disso, as WCAG 2.0 viraram um padrão ISO em 2012 (https://www.iso.org/standard/58625.html) e em 2014 foi o primeiro documento a obter uma tradução autorizada pelo W3C para português do Brasil (disponível em https://www.w3.org/Translations/WCAG20-pt-br/).

Os critérios de sucesso das WCAG 2.0 são as seguintes:

- **1.1.1 Conteúdo não textual (A)** - Orientações para alternativas a conteúdo não textuais, como imagens e campos de formulário.
- **1.2.1 Apenas áudio e apenas vídeo (pré-gravado) (A)** - Sobre alternativas para áudio e vídeo pré-gravado.
- **1.2.2 Legendas (pré-gravadas) (A)** - Fornecer legendas para conteúdo de áudio pré-gravado.

- **1.2.3 Audiodescrição ou mídia alternativa (pré-gravada) (A)** - Fornecer audiodescrição ou mídia alternativa para vídeo pré-gravado.
- **1.2.4 Legendas (ao vivo) (AA)** - Sobre adicionar legendas para conteúdo de áudio em vídeo ao vivo.
- **1.2.5 Audiodescrição (pré-gravada) (AA)** - Sobre fornecimento apenas de audiodescrição.
- **1.2.6 Língua de sinais (pré-gravada) (AAA)** - Disponibilização de interpretação de língua de sinais.
- **1.2.7 Audiodescrição estendida (pré-gravada) (AAA)** - Quando as pausas no vídeo são insuficientes para descrever o conteúdo.
- **1.2.8 Mídia alternativa (pré-gravada) (AAA)** - Alternativa para todo o conteúdo audiovisual.
- **1.2.9 Apenas áudio (ao vivo) (AAA)** - Alternativa para conteúdo apenas em áudio.
- **1.3.1 Informações e relações (A)** - Transmissão por código das relações e informações relativas ao conteúdo.
- **1.3.2 Sequência com significado (A)** - Quando a sequência de leitura altera o significado, alternativas são necessárias.
- **1.3.3 Características sensoriais (A)** - Garantir que a informação não depende de características sensoriais, como forma e cor.
- **1.4.1 Utilização de cores (A)** - Não utilizar a cor como única forma de transmitir informação.
- **1.4.2 Controle de áudio (A)** - Mecanismo para pausar ou interromper áudio que toca automaticamente.
- **1.4.3 Contraste (mínimo) (AA)** - O grau mínimo de contraste para evitar barreiras de acesso.
- **1.4.4 Redimensionar texto (AA)** - Possibilidade de

redimensionar texto sem comprometer o conteúdo ou visual.

- **1.4.5 Imagens de texto (AA)** - Como lidar com imagens que contenham texto.
- **1.4.6 Contraste (melhorado) (AAA)** - Grau ampliado de contraste.
- **1.4.7 Áudio de fundo baixo ou sem áudio de fundo (AAA)**- Como lidar com áudio de fundo/ som ambiente.
- **1.4.8 Apresentação visual (AAA)** - Como apresentar e lidar com blocos de texto.
- **1.4.9 Imagens de texto (sem exceção) (AAA)** - Imagens de texto para fins decorativos.
- **2.1.1 Teclado (A)** - Sobre como tornar o conteúdo acessível por teclado, com algumas exceções.
- **2.1.2 Sem bloqueio do teclado (A)** - Evitar barreiras de teclado (*keyboard trap*).
- **2.1.3 Teclado (sem exceção) (AAA)** - Garantir toda a navegação por teclado, sem exceções.
- **2.2.1 Ajustável por temporização (A)** - Como garantir acessibilidade com sistemas de tempo limite definido.
- **2.2.2 Colocar em pausa, parar, ocultar (A)** - Como lidar com conteúdo em movimento ou atualização automática.
- **2.2.3 Sem temporização (AAA)** - Evitar temporização.
- **2.2.4 Interrupções (AAA)** - Possibilidade de adiar ou suprimir interrupções.
- **2.2.5 Nova autenticação (AAA)** - O usuário não perde dados em uma nova autenticação.
- **2.3.1 Três flashes ou abaixo do limite (A)** - Evitar conteúdo que pisca ou utilizá-lo com limites de frequência e tamanho de tela.

- **2.3.2 Três flashes (AAA)** - Não utilizar conteúdo que pisca três vezes em menos de um segundo.
- **2.4.1 Ignorar blocos (A)** - Fornecer formas de saltar ou ignorar blocos de conteúdo.
- **2.4.2 Página com título (A)** - Páginas com título que descrevem sua função ou finalidade.
- **2.4.3 Ordem do foco (A)** - Sobre navegação e ordem de foco dos elementos navegados.
- **2.4.4 Finalidade do link (em contexto) (A)** - Determinar o significado e finalidade do link a partir do link sozinho (evitar o "clique aqui").
- **2.4.5 Várias formas (AA)** - Proporcionar várias formas para localizar conteúdo na página.
- **2.4.6 Cabeçalhos e rótulos (AA)** - Os cabeçalhos e os rótulos descrevem o tópico ou a finalidade.
- **2.4.7 Foco visível (AA)** - Elementos interativos tem o foco visível quando navegados por teclado.
- **2.4.8 Localização (AAA)** - Informações sobre a localização do usuário na página e site.
- **2.4.9 Finalidade do link (apenas o link) (AAA)** - mecanismo para determinar a finalidade do link quando esse não for claro ou ambíguo.
- **2.4.10 Cabeçalhos da seção (AAA)** - Os cabeçalhos da seção são utilizados para organizar o conteúdo.
- **3.1.1 Idioma da página (A)** - Definir o idioma da página.
- **3.1.2 Idioma das partes (AA)** - Definir o idioma de cada parte da página.
- **3.1.3 Palavras incomuns (AAA)** - Uma forma de ajudar o usuário a compreender palavras incomuns.
- **3.1.4 Abreviaturas (AAA)** - Disponibilizar um mecanismo

para identificar a forma expandida ou o significado das abreviaturas.

- **3.1.5 Nível de leitura (AAA)** - Alternativa para compreensão de um texto que exige nível de educação para sua compreensão.
- **3.1.6 Pronúncia (AAA)** - Proporcionar um mecanismo para determinar a pronúncia de palavras complexas ou ambíguas.
- **3.2.1 Em foco (A)** - Quando qualquer componente recebe o foco, não inicia uma alteração de contexto.
- **3.2.2 Em entrada (A)** - Alterar a definição de um componente de interface de usuário não provoca, automaticamente, uma alteração de contexto, a menos que o usuário tenha sido avisado sobre esse comportamento antes de utilizar o componente.
- **3.2.3 Navegação consistente (AA)** - Proporcionar uma navegação confortável e sem surpresas, a menos que o usuário seja notificado.
- **3.2.4 Identificação consistente (AA)** - Os componentes que têm a mesma funcionalidade em um conjunto de páginas web são identificados de forma consistente.
- **3.2.5 Alteração mediante solicitação (AAA)** - As alterações de contexto são iniciadas apenas a pedido do usuário.
- **3.3.1 Identificação do erro (A)** - Quando um erro acontece, o usuário deve ser notificado sobre como solucionar.
- **3.3.2 Rótulos ou instruções (A)** - Rótulos ou instruções são fornecidos quando o conteúdo exigir a entrada de dados por parte do usuário.

- **3.3.3 Sugestão de erro (AA)** - Sugestão sobre como solucionar um erro de entrada do usuário.
- **3.3.4 Prevenção de erros (legal, financeiro, dados) (AA)** - Como lidar com documentos e questões legais que envolvem intervenção do usuário.
- **3.3.5 Ajuda (AAA)** - Está disponível ajuda contextual.
- **3.3.6 Prevenção de erros (todos) (AAA)** - Se o usuário deve enviar informações pelo sistema, é possível disponibilizar uma forma de prevenção de erros conhecidos.
- **4.1.1 Análise (A)** - Desenvolver páginas e aplicações com base na especificação correta.
- **4.1.2 Nome, função, valor (A)** - Definir nome, função e valor das aplicações de forma acessível.

Os nomes de cada critério, na maioria dos casos, autoexplicativos. Porém, em alguns casos faz-se necessária a leitura detalhada do critério para a compreensão adequada.

As técnicas relacionadas a cada critério já foram listadas anteriormente e serão detalhadas no capítulo 5 desta publicação.

WCAG 2.1 (2018) - https://www.w3.org/TR/WCAG21/

Trata-se de uma pequena atualização da versão 2.0 que começou a trazer técnicas, diretrizes e critérios de sucesso relacionados a novas formas de acesso, deficiências cognitivas e de aprendizagem também (os critérios de sucesso são resultados diretos do trabalho do COGA - *Cognitive and Learning Disabilities Task Force*) além de contemplar dispositivos móveis.

As novas diretrizes ampliaram a cobertura das WCAG 2.0

trazendo os seguintes novos critérios de sucesso:

- **1.3.4 Orientação (AA)** - Referente à orientação horizontal/vertical do dispositivo.
- **1.3.5 Identificar o objetivo de entrada (AA)** - Definições do propósito de cada campo de formulário.
- **1.3.6 Identificar o objetivo (AAA)** - Propósito de cada elemento da página, seja de markup, ícones ou regiões interativas.
- **1.4.10 Realinhar (AA)** - Reorganização de conteúdo na página.
- **1.4.11 Contraste não textual (AA)** - Contraste para conteúdo não textual.
- **1.4.12 Espaçamento de texto (AA)** - Espaçamento de texto.
- **1.4.13 Conteúdo em foco por mouse ou teclado (AA)** - Comportamento referente a ações de passar o mouse e fazer foco por teclado.
- **2.1.4 Atalhos de teclado por caractere (A)** - Atalhos de teclado.
- **2.2.6 Limites de tempo (AAA)** - O que fazer quando uma aplicação necessita do uso de timeout.
- **2.3.3 Animação de interações (AAA)** - Sobre interatividade em animações.
- **2.5.1 Gestos de acionamento (A)** - Gestos de ponteiros.
- **2.5.2 Cancelamento de acionamento (A)** - Cancelamento de ponteiro (seja por mouse ou toque).
- **2.5.3 Rótulo em nome acessível (A)** - Rótulos relacionados ao nome.
- **2.5.4 Atuação em movimento (A)** - Como tratar acessibilidade de interfaces interativas por movimento.

- **2.5.5 Tamanho da área clicável (AAA)** - Tamanho do ponto de interatividade (toque ou clique).
- **2.5.6 Mecanismos de entrada simultâneos (AAA)** - Mecanismos de entrada concorrentes.
- **4.1.3 Mensagens de status (AA)** - Como tratar a acessibilidade de mensagens de status.

Até o momento, ainda não existe uma tradução autorizada deste documento, mas elas seguem a tradução livre do documento do W3C Brasil, que está disponível em: http://www.w3c.br/traducoes/wcag/wcag21-pt-BR/.

A maioria desses critérios de sucesso está contemplada nas técnicas para eliminar barreiras de acesso disponíveis no capítulo 5 desta publicação.

> **Importante**: apesar do breve resumo nesta seção e das técnicas apresentadas no capítulo 5, são fortemente recomendados a leitura, estudo e compreensão da documentação completa das WCAG. É um documento extremamente importante para a acessibilidade na Web.

3.2 WAI-ARIA

A documentação WAI-ARIA, que significa *Accessible Rich Internet Applications* (https://www.w3.org/TR/wai-aria-1.1/), foi criada com o objetivo de lidar com conteúdos dinâmicos em aplicações Web.

Conforme a Web evoluiu, as aplicações também começaram a utilizar recursos técnicos que iam além da simples semântica do HTML. A popularização de AJAX (Asynchronous JavaScript and XML) facilitou diversos processos em páginas Web, principalmente a possibilidade de atualização de parte do conteúdo da página sem a necessidade de recarregar todo o documento HTML. Com isso, surgiu a necessidade de identificar e tratar os eventos que ocorrem nas áreas da página.

É comum nos depararmos com aplicações que necessitem de mudança de semântica para funcionar. Em vez de utilizar os elementos interativos do HTML, algumas situações (principalmente em sistemas legados) nos levam a lidar com aplicações com elementos inadequados, como utilizar um `span` para fazer um botão. Mesmo assim, é possível fazê-lo se comportar como um botão, utilizando os atributos de ARIA:

```
<span role="button">Enviar</span>
```

Essa é uma representação simples de seu uso, já que fazer um elemento `span` se comportar como um botão envolve também uma boa dose de JavaScript.

> **Importante**: sempre que possível, prefira utilizar os elementos semânticos do HTML. Eles já vêm com seu peso semântico adequado definido.

A partir dessa necessidade, surgiu a documentação de WAI-ARIA, que oferece ontologias e regras (em forma de atributos)

para tornar as aplicações Web mais acessíveis. Esses atributos são divididos entre duas categorias: `roles` (papéis ou funções) e `states and properties` (estados e propriedades).

O atributo `role`

O atributo `role` é a representação semântica do seu papel (ou função). Um elemento com o atributo `role` tem por definição uma expectativa de comportamento com base na sua aplicação. Definir um elemento como `role=img` determina que sua semântica vai mudar para que ele se comporte como uma imagem, ou um `role=presentation` vai determinar que o elemento perdeu toda a semântica da sua aplicação.

Os papéis da aplicação são separados da seguinte forma:

Papéis abstratos (Abstract role)

Servem de conceito geral da taxonomia de WAI-ARIA. Costumam ser utilizados em *containers* específicos e áreas de aplicações. Não devem ser utilizadas para definir conteúdo:

- `command` - Widget que executa uma ação, mas não recebe dados de entrada.
- `composite` - Widget que pode conter elementos filhos que mantêm sua propriedade.
- `input` - Um tipo genérico de widget que permite a entrada de dados ou interação do usuário.
- `landmark` - Uma seção perceptível que contém conteúdo relevante para um propósito específico da aplicação ou página.
- `range` - Uma entrada de dados que representa um

intervalo de valores que pode ser definido pelo usuário.

- `roletype` - Função básica que todos os outros papéis nessa taxonomia herdam.
- `section` - Contêiner estrutural renderizável em um documento ou aplicação.
- `sectionhead` - Rotula ou resume sua respectiva seção.
- `select` - Widget de formulário que permite ao usuário fazer seleções a partir de um conjunto de opções.
- `structure` - Elemento contêiner da estrutura do documento.
- `widget` - Um componente interativo da interface do usuário.
- `window` - Um navegador ou aplicação janela.

Papéis de Widget (Widget roles)

Os atributos a seguir devem ser utilizados para definir aplicações de interface do usuário, independentes ou como partes de outras aplicações:

- `button` - Permite ações do usuário quando clicado ou pressionado.
- `checkbox` - Campo de entrada que tem valores selecionáveis.
- `gridcell` - Uma célula em uma grade.
- `link` - Uma referência interativa para um recurso interno ou externo que, quando ativado, faz com que o agente de usuário navegue para determinado recurso.
- `menuitem` - Conjunto de opções contidas em um menu ou barra de menu.
- `menuitemcheckbox` - Um item de menu com um estado

verificável.

- `menuitemradio` - Um item de menu verificável em um conjunto de elementos com o mesmo papel, dos quais apenas um pode ser verificado.
- `option` - Um item selecionável em uma lista de seleção.
- `progressbar` - Um elemento que exibe o status de progresso de tarefas que levam muito tempo.
- `radio` - Uma entrada verificável em um grupo de elementos com a mesma função, que apenas um pode ser acionado por vez.
- `scrollbar` - Um objeto gráfico que controla a rolagem de conteúdo dentro de uma área de visualização.
- `searchbox` - Um tipo de caixa de texto destinado a especificar critérios de busca.
- `separator` - Uma divisória que separa e diferencia seções de conteúdo ou grupos de itens de menu.
- `slider` - Uma entrada do usuário onde ele seleciona um valor a partir de um determinado intervalo.
- `spinbutton` - Uma forma de leque que espera que o usuário selecione uma entre opções distintas.
- `switch` - Um tipo de caixa que representa valores de ligado/desligado.
- `tab` - Uma etiqueta de agrupamento proporcionando um mecanismo para selecionar conteúdo.
- `tabpanel` - Um recipiente com um separador, em que cada aba está contida em uma `tablist` .
- `textbox` - Um tipo de entrada que permite texto livre como valor.
- `treeitem` - Um item de opção de uma árvore. Este é um elemento dentro de uma árvore que pode ser expandido ou

fechado se contém um grupo de subnível.

Os seguintes papéis atuam como widgets de interface de usuário compostos. Eles costumam atuar como contêineres que gerenciam outros widgets neles contidos.

- `combobox` - Um elemento contendo uma caixa de texto de linha única e um outro elemento em formato de lista ou grade contendo valores.
- `grid` - Um elemento contendo um conjunto de uma ou mais linhas, com uma ou mais células, que podem ser acessadas utilizando navegação por setas do teclado.
- `listbox` - Permite ao usuário selecionar um ou mais itens de uma lista de opções.
- `menu` - Widget que oferece uma lista de opções para o usuário.
- `menubar` - Menu que normalmente permanece visível e é normalmente apresentado na horizontal.
- `radiogroup` - Um grupo de botões de rádio.
- `tablist` - Lista de elementos de guia.
- `tree` - Um tipo de lista que pode conter grupos de subnível aninhados (podem ser reduzidos e expandidos).
- `treegrid` - Uma grade cujas linhas podem ser expandidas e recolhidas em formato de galhos de árvore.

Estrutura de documento (Document Structure)

Os atributos a seguir servem para descrever estruturas que organizam o conteúdo em uma página. Essas estruturas de documentos geralmente não são interativas.

- `application` - Uma estrutura que contém um ou mais

elementos que necessitam ser focados pela entrada do usuário.

- `article` - Elemento que forma uma parte independente de um documento, página ou site.
- `cell` - Uma célula em um recipiente tabular.
- `columnheader` - Célula contendo informação do cabeçalho de uma coluna.
- `definition` - A definição de um termo ou conceito.
- `directory` - Uma lista de referências, como uma tabela estática de conteúdos.
- `document` - Um elemento de conteúdo.
- `feed` - Uma lista deslocável de artigos onde é possível adicionar ou remover itens de uma ou outra extremidade da lista.
- `figure` - Uma seção perceptível de conteúdo que normalmente contém um documento gráfico, imagens, trechos de código, ou um exemplo de texto.
- `group` - Um conjunto de objetos.
- `heading` - O cabeçalho de uma seção da página.
- `img` - Um recipiente para um conjunto de elementos que formam uma imagem.
- `list` - Uma seção que contém elementos de lista.
- `listitem` - Um único item em uma lista ou diretório.
- `math` - Conteúdo que representa uma expressão matemática.
- `none` - Um elemento cuja semântica nativa não será mapeada para a API de acessibilidade.
- `note` - Uma seção cujo conteúdo é acessório ao conteúdo principal do recurso.
- `presentation` - Um elemento cuja semântica nativa não

será mapeada para a API de acessibilidade. Utilizar este atributo tira toda a semântica do elemento.

- `row` - Uma linha de células em um recipiente tabular.
- `rowgroup` - Uma estrutura contendo uma ou mais linhas de elementos.
- `rowheader` - Uma célula que contém informações de cabeçalho da linha.
- `separator` - Uma divisória que separa e diferencia seções de conteúdo ou grupos de itens de menu.
- `table` - Uma seção que contém os dados organizados em linhas e colunas.
- `term` - Uma palavra ou frase que possui uma definição correspondente.
- `toolbar` - Conjunto de botões de função usados para representar controles de forma compacta.
- `tooltip` - Um pop-up contextual que exibe uma descrição de um elemento.

Papéis de marcação (Landmark roles)

Representam áreas da página ou aplicação. Com a popularização dos elementos semânticos de HTML5 boa parte desses atributos começam a se tornar desnecessários.

- `banner` - Uma região de conteúdo normalmente posicionada no topo da página.
- `complementary` - A seção de suporte do documento, concebido para ser complementar ao conteúdo principal a um nível semelhante na hierarquia do DOM, mas continua a ser significativa quando separado do conteúdo principal.
- `contentinfo` - A grande região perceptível que contém

informações relacionadas ao documento pai.

- `form` - A região macro que contém uma coleção de itens e objetos que, como um todo, se combinam para criar um formulário.
- `main` - O conteúdo principal de um documento.
- `navigation` - Uma coleção de elementos (normalmente links) para navegar pelo documento ou em documentos relacionados.
- `region` - Uma seção de conteúdo para uma finalidade específica.
- `search` - A região que contém uma coleção de itens e objetos que, como um todo, se combinam para criar um mecanismo de busca.

Papéis de regiões ativas (Live region roles)

Representam as regiões ativas e que podem ser modificadas por atributos. Normalmente são regiões que são atualizadas dinamicamente e devem transmitir as mudanças ao usuário.

- `alert` - Um tipo de região com informações importantes e geralmente urgentes.
- `log` - Um tipo de região onde novas informações são adicionadas em ordem significativa e informações antigas podem desaparecer.
- `marquee` - Um tipo de região onde a informação não essencial muda frequentemente.
- `status` - Um tipo de região cujo conteúdo é a informação exibida para o usuário, mas não é importante o suficiente para justificar um alerta.
- `timer` - Um tipo de região contendo um contador

numérico que indica uma quantidade de tempo decorrido desde um ponto de partida ou o tempo restante até um ponto final.

Papéis de janela (Window roles)

Agem como janelas dentro do navegador ou aplicativo.

- `alertdialog` - Contém uma mensagem de alerta, onde o foco inicial vai para um elemento dentro da caixa de diálogo.
- `dialog` - Uma caixa de diálogo é uma janela descendente da janela principal de uma aplicação web.

Essa lista foi feita e traduzida de forma livre com base na lista detalhada com a função de cada uma das `roles`, disponível no site do W3C, em https://www.w3.org/TR/wai-aria-1.1/#role_definitions/

Estados e propriedades

São uma série de atributos com valores relacionados aos papéis definidos pelo atributo `role`. Eles fornecem informações que podem ser transmitidas para o usuário a qualquer momento pela sua tecnologia assistiva. E isso pode acontecer em tempo real ou predefinida pelo autor com base em eventos e ações do usuário.

Da mesma forma que os papéis são separados por tipos de atributos, o mesmo serve para os estados e propriedades. Cada tipo de atributo serve para definir ações de acordo com o seu tipo de utilização.

São eles:

Atributos de Widget (Widget Attributes)

Atributos específicos para elementos comuns da interface do usuário. São utilizados principalmente para suportar ações interativas e para suportar as funções do widget.

- `aria-autocomplete` - Indica se a introdução de texto poderia desencadear exibição de valores já previstos.
- `aria-checked` - Indica o estado "marcado" de caixas de seleção, botões de rádio, e outros widgets.
- `aria-disabled` - Indica que o elemento é perceptível, mas desativado, por isso não é editável ou operável.
- `aria-errormessage` - Identifica o elemento que fornece uma mensagem de erro para o objeto.
- `aria-expanded` - Indica se o elemento, ou outro elemento de agrupamento que controla, está expandido ou fechado.
- `aria-haspopup` - Indica a disponibilidade e o tipo de elemento de contexto interativo.
- `aria-hidden` - Indica se o elemento é exposto a uma API de acessibilidade.
- `aria-invalid` - Indica se o valor inserido não está de acordo com o formato esperado pela aplicação.
- `aria-label` - Define um valor que rotula o elemento atual.
- `aria-level` - Define o nível hierárquico de um elemento dentro de uma estrutura.
- `aria-modal` - Indica se um elemento é modal quando exibido.
- `aria-multiline` - Indica se uma caixa de texto aceita múltiplas linhas de entrada ou apenas uma única linha.

- `aria-multiselectable` - Indica que o usuário pode selecionar mais de um item a partir dos descendentes selecionáveis.
- `aria-orientation` - Indica se a orientação do elemento é horizontal, vertical, ou desconhecida/ambígua.
- `aria-placeholder` - Define uma pequena dica (uma palavra ou frase curta) destinada a auxiliar o usuário com a entrada de dados quando o controle não tem valor. Uma dica pode ser um valor de amostra ou uma breve descrição do formato esperado.
- `aria-pressed` - Indica o estado "pressionado" de botões de alternância.
- `aria-readonly` - Indica que o elemento não é editável, mas é operável.
- `aria-required` - Indica que a entrada do usuário é necessária no elemento antes de um formulário ser enviado.
- `aria-selected` - Indica o estado "selecionado" de widgets.
- `aria-sort` - Indica se os itens em uma tabela ou grade são classificadas em ordem crescente ou decrescente.
- `aria-valuemax` - Define o valor máximo permitido para um widget de intervalo.
- `aria-valuemin` - Define o valor mínimo permitido para um widget de intervalo.
- `aria-valuenow` - Define o valor atual para um widget de intervalo.
- `aria-valuetext` - Define o texto alternativo legível por humanos.

Atributos de regiões ativas (Live Region Attributes)

O objetivo desses atributos é indicar que as alterações de conteúdo podem ocorrer sem o elemento ter foco e fornecer informações para as tecnologias assistivas auxiliarem no processo de atualizações de conteúdo.

- `aria-atomic` - Indica se as tecnologias assistivas vão apresentar tudo ou apenas partes da região que foi atualizada.
- `aria-busy` - Indica se um elemento está sendo modificado e que as tecnologias assistivas podem esperar até que as modificações sejam concluídas até que sejam expostos para o usuário.
- `aria-live` - Indica que um elemento será atualizado e descreve os tipos de atualizações que o agente de usuário, tecnologias assistivas e usuário podem esperar da região atualizada.
- `aria-relevant` - Indica quais notificações do agente de usuário vão disparar quando uma região dinâmica é modificada.

Atributos de relacionamento (Relationship Attributes)

Lista de atributos que indicam relacionamentos ou associações entre elementos que não podem ser prontamente determinados a partir da estrutura do documento.

- `aria-activedescendant` - Identifica o elemento atualmente ativo quando DOM recebe foco sobre um widget, caixa de texto, grupo ou aplicação.
- `aria-colcount` - Define o número total de colunas em

uma tabela ou grade.

- `aria-colindex` - Define índice de coluna de um elemento ou a posição em relação ao número total de colunas dentro de uma tabela ou grade.
- `aria-colspan` - Define o número de colunas ocupadas por uma célula ou `gridcell` dentro de uma tabela ou grade.
- `aria-controls` - Identifica o elemento (ou elementos), cujo conteúdo ou presença são controlados pelo elemento atual.
- `aria-describedby` - Identifica o elemento (ou elementos) que descreve outro objeto.
- `aria-details` - Identifica o elemento que fornece uma descrição detalhada estendida para o objeto.
- `aria-errormessage` - Identifica o elemento que fornece uma mensagem de erro para o objeto.
- `aria-flowto` - Identifica o próximo elemento (ou elementos) em uma ordem de leitura alternativa de conteúdo que, a critério do usuário, permite que tecnologia assistiva substitua o padrão geral de leitura do documento.
- `aria-labelledby` - Identifica o elemento (ou elementos) que rotula o elemento atual.
- `aria-owns` - Identifica um elemento (ou elementos), a fim de definir uma relação quando a hierarquia DOM não pode ser utilizada para representar a relação.
- `aria-posinset` - Define o número ou a posição de um elemento no conjunto atual de listas. Não é necessário se todos os elementos do conjunto estão presentes no DOM.
- `aria-rowcount` - Define o número total de linhas em uma tabela ou grade.

- `aria-rowindex` - Define índice de linha de um elemento ou a posição em relação ao número total de linhas dentro de uma tabela ou grade.
- `aria-rowspan` - Define o número de linhas ocupadas por uma célula ou `gridcell` dentro de uma tabela ou grade.
- `aria-setsize` - Define o número de itens no conjunto atual de listas. Não é necessário se todos os elementos do conjunto estão presentes no DOM.

Esta é uma lista dos atributos de WAI-ARIA da documentação do W3C na versão 1.1, e foi traduzida de forma livre para esta publicação. A lista original está disponível em https://www.w3.org/WAI/PF/aria-1.1/states_and_properties/.

A utilização de alguns destes atributos em casos comuns será explicada mais detalhadamente no capítulo 5, quando veremos como solucionar barreiras de acessibilidade utilizando os recursos referentes a cada uma das documentações.

3.3 ATAG

Sigla de *Authoring Tool Accessibility Guidelines*, a documentação ATAG (https://www.w3.org/TR/ATAG10/) tem como objetivo oferecer documentação para a criação de ferramentas de autoria que gerem conteúdo acessível e que seja acessível para as pessoas com deficiência.

É uma documentação muito específica, mas que tem suas particularidades, já que uma ferramenta de autoria pode ser um software de edição de HTML ou uma aplicação na Web que permite a criação de websites direto do navegador. São apenas sete

diretrizes, mas com recomendações importantes para quem vai criar uma aplicação que permita a produção de conteúdos Web. As diretrizes são as seguintes:

- **Diretriz 1.** Apoiar práticas de criação acessíveis.
- **Diretriz 2.** Gerar marcação conforme o padrão.
- **Diretriz 3.** Apoiar a criação de conteúdo acessível.
- **Diretriz 4.** Fornecer maneiras de verificar e corrigir o conteúdo inacessível.
- **Diretriz 5.** Integrar as soluções de acessibilidade na aparência geral.
- **Diretriz 6.** Promover acessibilidade na ajuda e documentação.
- **Diretriz 7.** Garantir que a ferramenta de autoria seja acessível a autores com deficiências.

Da mesma forma que as outras documentações possuem outros documentos de apoio, o documento ATAG também tem links para material de suporte, com recursos técnicos para a implementação e solução de problemas.

3.4 UAAG

Talvez este seja um documento muito específico com pouca relação com os demais, mas vale a pena apresentar ao público deste livro. O documento *User Agent Accessibility Guidelines* (https://www.w3.org/TR/UAAG/) disponibiliza diretrizes para quem vai desenvolver agentes de usuários (como navegadores) ou tecnologia assistiva para o acesso à Web.

Este documento anda em conjunto com o WCAG, pois

quando o desenvolvedor publica um código seguindo os padrões para páginas, os navegadores e tecnologia assistiva que seguem as UAAG interpretam o código de forma a não criar barreiras de acesso. Resumindo: as WCAG estão para o desenvolvedor Web, enquanto o UAAG está para o navegador e tecnologia assistiva.

3.5 WCAG-EM

Sigla de *Website Accessibility Conformance Evaluation Methodology* (https://www.w3.org/TR/WCAG-EM/) em tradução livre, Metodologia de avaliação de conformidade de acessibilidade do site. Este documento fornece orientação sobre a avaliação de conformidade com as Diretrizes de acessibilidade de conteúdo da Web (WCAG) 2.0. Ele descreve um procedimento para avaliar sites e inclui considerações para orientar os avaliadores e promover boas práticas. Não é uma ferramenta de validação e sim um guia para orientar os responsáveis por auditar a acessibilidade das páginas Web. No capítulo sobre conformidade e questões legais será apresentada uma ferramenta que auxilia o processo de verificação de acessibilidade com base no WCAG-EM.

3.6 EMAG

O Modelo de Acessibilidade de Governo Eletrônico (eMag - http://emag.governoeletronico.gov.br/) teve sua primeira versão lançada em 2005. A versão 3.1 foi lançada em 2014 e é a mais recente até o momento (dezembro de 2019).

Ele surgiu da falta de acessibilidade dos sites governamentais brasileiros, que tinham sido "enquadrados" do artigo 47 do

Decreto Nº 5.296 de 2 de dezembro de 2004, que dizia que:

"No prazo de até doze meses a contar da data de publicação deste Decreto, será obrigatória a acessibilidade nos portais e sítios eletrônicos da administração pública na rede mundial de computadores (internet), para o uso das pessoas portadoras de deficiência visual, garantindo-lhes o pleno acesso às informações disponíveis."

Como não existiam orientações sobre "como" tornar esses sites acessíveis, o eMag surgiu como um guia para tornar os sites públicos acessíveis.

De forma mais simples e de leitura descomplicada (diferente dos documentos do W3C) o eMag facilita a compreensão das diretrizes de acessibilidade. Ele foi construído com base nas diretrizes internacionais do W3C, mas diferente das recomendações internacionais; não existem níveis de adoção: para os sites governamentais, tudo é obrigatório.

As recomendações, 45 no total, são divididas da seguinte forma:

- Marcação
- Comportamento (DOM)
- Conteúdo/Informação
- Apresentação/Design
- Multimídia
- Formulário

Dentro de cada recomendação é possível identificar a relação entre a documentação do W3C e até exemplos práticos de código.

3.7 SECTION 508

Em agosto de 1998, o então presidente dos Estados Unidos da América, Bill Clinton, assinou emendas da Lei de Reabilitação de 1998, que abrange o acesso a programas e serviços financiados pelo governo federal. A lei exige e garante o acesso à tecnologia pelo Governo Federal Americano.

A lei obriga que todos os sistemas sejam acessíveis para funcionários do governo e para a população em geral, garantindo que pessoas com deficiência não tenham barreiras de acesso aos serviços governamentais.

Como é uma lei que aborda "acessibilidade digital" e não somente a Web, foi criado um site (https://www.section508.gov/) para agrupar as informações referentes às técnicas necessárias para cumprir a Section 508.

Na seção destinada à criação de Websites acessíveis, as orientações do material disponível são bem diretas. A lista a seguir é uma tradução livre do conteúdo em https://www.section508.gov/create/software-websites/:

- Encontre treinamento de acessibilidade orientado ao desenvolvedor;
- Entenda como os padrões relacionados a 508 se aplicam ao conteúdo, sistemas, plataformas e estruturas eletrônicas;
- Siga as melhores práticas da ACOP (Federal Accessibility Community of Practice);
- Reveja e priorize as melhores práticas de criação do W3C, WCAG e ARIA;
- Use ferramentas automatizadas, testes manuais e testes

baseados em tecnologia assistiva;
- Examine as melhores práticas para aplicativos móveis e ambientes da Web; e
- Implemente estruturas de desenvolvimento populares.

3.8 ADA

O Departamento de Justiça Americano publicou em dezembro de 2010 uma série de padrões com o objetivo de garantir a acessibilidade em diversas áreas. É um documento muito amplo que aborda desce acessibilidade em piscinas e venda de ingressos até serviços online de Websites. Esse documento é o ADA, sigla de American Disabilities Act (https://www.ada.gov/).

A parte relacionada à acessibilidade na Web é um documento (https://www.ada.gov/websites2.htm) de aproximadamente cinco páginas com as orientações para estar de acordo com a ADA. Ele menciona a Section 508 e as diretrizes do W3C para especificações técnicas.

3.9 OUTRAS DIRETRIZES TÉCNICAS IMPORTANTES

As diretrizes mencionadas anteriormente talvez sejam as mais conhecidas, mas existe uma série de outros documentos que abordam a questão da acessibilidade digital com viés mais específico. Isso não significa que eles sejam menos importantes que os demais.

Conheça alguns deles a seguir:

COGA

Sigla de *Cognitive and Learning Disabilities Accessibility*, o documento aborda técnicas de acessibilidade na Web para contemplar o acesso de pessoas com deficiências neurológicas, limitações cognitivas e dificuldades de aprendizado. A versão de janeiro de 2019 tem seu status de W3C Editor's Draft, ou seja, ainda é um documento inicial que deve evoluir. Está disponível em https://w3c.github.io/coga/techniques/.

Em português, existe um guia de recomendações para desenvolvimento de sites mais acessíveis a pessoas com Autismo (GAIA) desenvolvido pela Talita Pagani, disponível em http://talitapagani.com/gaia/sobre/

TTML

O padrão para legendagem de vídeo na Web está em sua versão 2.0. O documento Timed Text Markup Language traz orientações para a criação de legendas acessíveis para conteúdo audiovisual publicado na rede. Está disponível em https://www.w3.org/TR/2018/REC-ttml2-20181108/.

Mobile Accessibility

Área no site do próprio W3C para agrupar diretrizes de acessibilidade direcionadas para dispositivos móveis. Mais uma vez o W3C não quer reinventar a roda e identifica dentro dos padrões consolidados as técnicas para tornar conteúdo acessível quando o usuário estiver utilizando dispositivos móveis. Está disponível em https://www.w3.org/WAI/standards-guidelines/mobile/.

ACT

Sigla de *Accessibility Conformance Testing*, o documento define regras e técnicas para testes de acessibilidade em aplicações Web, sejam eles testes totalmente automatizados, semiautomatizados ou totalmente manual. O documento tem como objetivo permitir a transparência e a harmonização dos métodos de teste, incluindo métodos implementados por ferramentas de teste de acessibilidade. Está disponível em https://www.w3.org/TR/act-rules-format/.

EARL

O Evaluation and Report Language (EARL) é um formato legível por máquina para expressar os resultados dos testes de acessibilidade em páginas e sistemas Web. A principal motivação para o desenvolvimento do EARL é facilitar o processamento de resultados de testes, como aqueles gerados pelas ferramentas de avaliação de acessibilidade da Web, usando um formato independente de fornecedor e independente de plataforma. Está disponível em https://www.w3.org/WAI/standards-guidelines/earl/.

Game Accessibility guidelines

Não são diretrizes do W3C muito menos tocam diretamente em tópicos relacionados especificamente a jogos na Web, mas traz boas dicas para o desenvolvimento de jogos online para que não crie barreiras de acesso para pessoas com deficiência. Está disponível em http://gameaccessibilityguidelines.com/.

Capítulo 4

ELIMINANDO AS PRINCIPAIS BARREIRAS DE ACESSO

Esta talvez seja uma das partes mais importantes desta publicação. A partir deste capítulo vamos nos aprofundar nas técnicas para eliminar barreiras de acesso em páginas Web. Abordaremos desde as mais simples intervenções no código, como links acessíveis, até aplicações mais complexas como conteúdo dinâmico e multimídia.

Nos capítulos dedicados às técnicas e exemplos de código optei por lidar com as diretrizes de forma a "eliminar barreiras de acesso". Ao fazer isso, estamos tratando a questão da acessibilidade de forma tangível. O objetivo de tratar a parte técnica dessa forma (e não descrevendo cada uma das diretrizes do WCAG) é tornar a compreensão muito mais simples e prática para a implementação imediata.

Com essa nova forma de apresentar esse conteúdo você vai perceber que determinadas barreiras dependem de combinações de técnicas para serem solucionadas. Por isso, a abordagem diferente. Isso vai facilitar a vida de quem quer tornar o conteúdo

acessível, mas o documento WCAG 2.1 continua como referência detalhada. Quero que este capítulo funcione como um esclarecimento de dúvidas em vez de ser uma lista de requisitos que precisam ser atingidos.

Antes de começar, vale relembrar que o objetivo deste livro é eliminar o máximo de barreiras de acesso possíveis. Dificilmente um site ou aplicação será 100% acessível, dadas as peculiaridades de deficiências específicas, mas a eliminação de barreiras é tangível e mensurável, como veremos ao final desta publicação.

As técnicas para eliminar barreiras de acesso serão divididas em tópicos. Serão abordadas técnicas de acessibilidade considerando as seguintes áreas:

4.1 ESTRUTURA

Consiste na parte de organização semântica e hierárquica do conteúdo de uma página. Essa estrutura precisa ser robusta para que o conteúdo tenha significado e a aplicação não seja um monte de código jogado dentro de uma página HTML.

Uma estrutura bem definida permite que tecnologia assistiva (e outras aplicações) consiga identificar as principais áreas da aplicação e transmitir essa informação para o usuário, seja ela uma informação estática ou dinâmica.

Considerar uma estrutura acessível atinge diretamente o quarto princípio das WCAG, que é tornar a aplicação robusta. Uma aplicação robusta necessita de menos código e mais informação com significado. Menos `containers` genéricos representa uma aplicação bem estruturada e mais acessível.

4.2 NAVEGAÇÃO E INTERATIVIDADE

A Web não é somente uma aplicação para um expectador passivo. Ela é dinâmica e construída com base na interação do usuário, que pode mudar completamente uma interface com simples toques em botões. A interatividade está na essência da Web e ela precisa ser acessível para todas as pessoas.

Interatividade não significa que uma pessoa vai acessar um link ou botão com toque em uma tela ou com um mouse. Esse usuário pode acionar um controle interativo sem as mãos, por exemplo. E pode acessar das mais variadas formas, seja com uma ponteira na cabeça ou com um comando de voz.

O tópico sobre navegação e interatividade aborda os principais recursos relacionados à interação do usuário, como conteúdos interativos de links e ações do usuário em botões e aplicação.

4.3 DESIGN

A preocupação com a acessibilidade de uma aplicação vai além do código para garantir que a percepção do usuário seja adequada e que barreiras sensoriais não interfiram na forma como o usuário consome conteúdo na página. O design deve considerar a experiência do usuário e as suas percepções de formas, cores e posições de elementos em uma página.

Quando digo que o design vai além do código significa que a acessibilidade vem muito antes de transformar um wireframe em um website. As preocupações com design acessível devem vir da reunião de projeto, da escolha de cores e da concepção artística da aplicação.

Pensar em design acessível depois de uma aplicação pronta é um risco muito grande, já que pode interferir nos tamanhos de textos e objetos e em cores e contrastes da aplicação.

4.4 MULTIMÍDIA E CONTEÚDO NÃO TEXTUAL

Nos primórdios da Web nunca se imaginava que aplicações de áudio e vídeo pudessem fazer parte corriqueira do nosso dia a dia na rede. Elas estão, e cada vez mais fortes e presentes. Estamos presenciando um momento do audiovisual tomando o lugar do texto em alguns cenários. E para isso esse material precisa ser acessível.

Mas quando falamos em conteúdo não textual, não estamos falando apenas de áudio e vídeo. Precisamos falar de imagens acessíveis. Que pessoas que não enxergam as imagens possam compreender o que ela representa e significa.

Tornar esse conteúdo acessível significa garantir que uma pessoa com deficiência vai conseguir compreender uma imagem, um vídeo ou um podcast em um website. Existem técnicas que possibilitam a acessibilidade desses recursos, e é isso que veremos na seção de multimídia e conteúdo não textual.

Sendo assim, vamos para a eliminação das barreiras de acesso.

CAPÍTULO 5

ELIMINANDO BARREIRAS - ESTRUTURA

A estrutura de uma página ou aplicação na Web precisa ser robusta o suficiente para que tanto o usuário quanto ferramentas de busca ou navegadores consigam compreender o significado das informações em cada parte da página. O peso semântico de cada elemento é fundamental para que navegadores e leitores de tela possam compreender e transmitir a informação de forma adequada.

Esse peso semântico ganhou força no HTML5, pois antes disso (considerando HTML4 ou XHTML1) eram utilizados `containers` genéricos para elementos estruturais, o que fazia com que informações de rodapé pudessem ter o mesmo peso semântico de informações em áreas centrais da aplicação.

Antigamente utilizávamos `containers` como `<div>` para qualquer coisa, e com os elementos semânticos do HTML5 isso mudou. Isso não significa que não podemos mais usar elementos genéricos, mas que podemos dar peso semântico. Usar `<footer>` para um rodapé é muito melhor do que utilizar um elemento `<div>`.

Outro bom exemplo disso são os campos de formulário. Quando entregamos para o usuário uma aplicação com um simples `<input type="text" name="telefone">` não estamos usufruindo de todo o potencial que a linguagem de marcação nos proporciona. Mas quando usamos `<input type="tel" name="telefone">` para representar um campo que exige dados de um número de telefone estamos transmitindo para todos os atores que aquele campo tem um significado.

Dessa forma não só os agentes de usuário como tecnologia assistiva podem entregar para o usuário uma experiência muito mais completa, principalmente com relação à acessibilidade.

Isso é o que vamos abordar neste capítulo. Conhecer diversas técnicas para dar significado ao conteúdo na Web.

5.1 TÍTULO DA PÁGINA

É a primeira informação carregada no navegador e transmitida para o usuário de tecnologia assistiva, principalmente para leitores de tela. A informação dentro do elemento `<title>` precisa ser clara não só para o usuário quanto para ferramentas de busca que indexam esse conteúdo.

A sintaxe é simples:

```
<title>Blog de Reinaldo Ferraz</title>
```

Porém, esse recurso requer cuidado. Utilizar o mesmo título para todas as páginas é ruim, pois não transmite a informação de qual a página em que o usuário acabou de entrar. Também não é uma boa prática colocar a área do site depois do nome.

Em vez de fazer isso:

```
<title>Blog de Reinaldo Ferraz | Sobre</title>
```

Faça isso:

```
<title>Sobre | Blog de Reinaldo Ferraz</title>`
```

Isso facilita o acesso por usuários que não conseguem enxergar a página inteira e possibilita que eles saibam do que significa essa página com base na primeira informação transmitida pelo leitor de tela.

Essa técnica de colocar a área do site antes do nome também ajuda na localização do conteúdo por ferramentas de busca. É muito mais fácil ler as primeiras informações do resultado do que procurar em cada link essa informação.

Finalmente... um blog | Reinaldo Ferraz
www.reinaldoferraz.com.br/finalmente-um-blog/
13 de nov de 2013 - Relutei e demorei bastante para criar um blog, mas acho que agora chegou a hora. Depois de escrever sobre publicar conteúdo aberto na ...
Reinaldo Ferraz | Open Web, Acessibilidade, Star Wars e Café
www.reinaldoferraz.com.br/
Acessibilidade na Web - Reinaldo Ferraz ..... Eu havia esquecido de configurar o robots.txt do blog e por isso demorou um pouco para a indexação do conteúdo ...
Sobre | Reinaldo Ferraz
www.reinaldoferraz.com.br/sobre/
Foto de Reinaldo Ferraz. Formado em desenho e ... Minha lista de artigos no blog está desat...
https://t.co/m6UuSUpfBz · 11 horas ago; Fica a dica para o ...

Figura 5.1: Imagem dos resultados de busca do Google. A imagem mostra três resultados da busca por blog Reinaldo Ferraz e os resultados aparecem com a área do site antes do nome.

Sites de notícia também fazem um bom uso dessa técnica, pois o título da notícia, que é algo extremamente relevante, aparece

como a primeira informação na busca por seu conteúdo.

Quando não se usa essa ordem, duas dificuldades podem ser listadas: os leitores de tela sempre vão repetir o nome do site/instituição primeiro e ferramentas de busca podem não conseguir exibir textos longos. O exemplo a seguir ilustra essa situação:

Sítio da Câmara Municipal de Lisboa: Participação de ocorrência em ...
www.cm-lisboa.pt › Serviços › Pedidos › Pagamentos, taxas e tarifas
Pode fazer o pedido à Câmara Municipal de Lisboa através de correio eletrónico, correio postal ou presencialmente nas nossas lojas de atendimento, mediante ...
Sítio da Câmara Municipal de Lisboa: galeria
www.cm-lisboa.pt/galeria
Galeria. Galeria-Lisboa · Eixo Central. Cidade. 120 Imagens. Torre de Belém. Museus e Patrimonio. 100 Imagens. Parque das Nações. Lazer e Entretenimento.
Sítio da Câmara Municipal de Lisboa: Urbanismo e obras
www.cm-lisboa.pt › Serviços › Pedidos
Selecione o tema que procura e saiba como e onde pode tratar dos seus pedidos. Consulte aqui as perguntas frequentes sobre este assunto. Autorizações de ...

Figura 5.2: Imagem dos resultados de busca do Google. A imagem mostra três resultados da busca por Sítio da Câmara Municipal de Lisboa, onde os resultados aparecem depois do nome.

Qual a barreira de acesso? O elemento `title` serve para dar a primeira informação sobre o que o usuário encontrará na página. Ele precisa ser direto e claro, pois sem isso usuários de software leitores de tela não conseguirão compreender que página acessaram sem navegar por toda ela.

5.2 ESTRUTURA SEMÂNTICA

Antes do HTML5, a Web era dominada pelos elementos `<div>` . Por ser um `container` genérico, ele facilitou e flexibilizou o desenvolvimento de aplicações, mas deixou a semântica e o significado de cada bloco de lado. Com o HTML5, isso ficou mais fácil, pois existem `containers` específicos para cada tipo de necessidade. Os mais comuns para estruturar uma página Web são os seguintes:

- `<header>` para agrupar elementos no topo da página;
- `<nav>` para definir uma área de navegação;
- `<section>` define uma seção genérica da aplicação;
- `<article>` define uma seção de agrupamento de conteúdo;
- `<aside>` serve para apresentar conteúdo complementar;
- `<footer>` para informações de rodapé de página;
- `<h1>-<h6>` cabeçalhos da página;
- `<p>` parágrafos;
- `<br>` quebra de linha;
- `<img>` para aplicação de uma imagem;
- `<figure>` para agrupar elementos de uma imagem;
- `<figcaption>` para colocar legendas em imagens;
- `<strong>` e `<em>` dão ênfase ao conteúdo. O primeiro como negrito e o segundo como itálico.

Diferente dos elementos `<b>` e `<i>` , que apresentam o texto respectivamente em negrito e itálico sem peso semântico, os elementos `<strong>` e `<em>` podem apresentar o conteúdo para usuários de leitores de tela com diferentes entonações para seus usuários, devido ao seu peso semântico.

Essa é uma pequena lista para ilustrar o quanto de significado cada elemento traz para a aplicação. A lista completa de elementos do HTML5 está disponível no site do W3C (https://www.w3.org/TR/html52/dom.html#elements).

Existe uma série de atributos de WAI-ARIA que servem para estender a semântica dos elementos, utilizando o atributo `role`. Valores como `role=banner` e `role=navigation` servem para marcar áreas equivalentes aos elementos `header` e `nav`, respectivamente, quando são utilizados como `containers` genéricos, por exemplo, utilizando um elemento `div`. Quando utilizados os elementos estruturais do HTML5 os atributos de `landmarks` tornam-se dispensáveis.

Em casos onde fica difícil mudar a semântica de elementos (como em sistemas legados, por exemplo), existe a possibilidade de adicionar atributos de WAI-ARIA para garantir a semântica da estrutura. Por exemplo, em um sistema que trata um botão como um elemento genérico pode utilizar atributos de WAI-ARIA para que ele se comporte como um botão (`<span role=buttom>Enviar</span>`), uma imagem (`<div role=img> </div>`) ou mesmo uma aplicação mais complexa, como um slider (`<div role=slider></div>`) ou uma barra de progresso (`div role="progressbar></div>`).

A lista completa com os valores do atributo `role` e suas respectivas funções foram detalhadas no capítulo anterior desta publicação.

O uso de atributos de WAI-ARIA deve ser feito de forma a identificar as áreas e widgets/aplicações da página.

Estrutura semântica para campos de formulário

A documentação do HTML5 também trouxe significado para campos de formulário. No passado utilizávamos `input type=text` para todo o tipo de informação, fosse um campo para preenchimento de e-mail ou de busca. No HTML5.2 temos à disposição o seguinte conjunto de atributos para uso em formulários:

Elementos do HTML5.2 herdados do HTML4:

- `input type=text` - para uma string simples e genérica de texto alfanumérico;
- `input type=hidden` - para uma string simples e genérica de texto alfanumérico que fica escondida;
- `input type=password` - para preenchimento de senha;
- `input type=checkbox` - para campos checkbox;
- `input type=radio` - para campos radiobox;
- `input type=file` - para o upload de arquivos externos;
- `input type=submit` - botão para envio do formulário;
- `input type=image` - para uso de imagem para o botão de envio do formulário;
- `input type=reset` - para apagar o formulário;
- `input type=button` - botão genérico.

Novos elementos disponíveis a partir do HTML5:

- `input type=search` - para campos de busca;
- `input type=tel` - para campo numérico de telefone;
- `input type=email` - para preenchimento de e-mail;
- `input type=url` - para preenchimento de um endereço na Web;

- `input type=date` - para preenchimento de data;
- `input type=month` - para preenchimento de mês;
- `input type=week` - para preenchimento de semana;
- `input type=time` - para preenchimento de horário;
- `input type=datetime-local` - para preenchimento horário em determinado timezone;
- `input type=number` - campo numérico;
- `input type=range` - valor numérico para uso de slider;
- `input type=color` - para criar um picker de cor.

Os novos atributos ajudam a ampliar o significado dos campos de formulário. No capítulo sobre navegação e interatividade mostrarei exemplos de como o uso desses novos recursos pode ser bom não só para acessibilidade quando para usabilidade.

Nem sempre todos os recursos podem ser acessíveis em determinada combinação de navegador *versus* tecnologia assistiva. Para saber se os elementos estruturais, formulários e atributos são acessíveis à API de acessibilidade dos navegadores, consulte a página https://www.html5accessibility.com/. Ela verifica como anda a implementação dos navegadores com relação a cada elemento e atributo referente a acessibilidade.

> **Qual a barreira de acesso?** Tecnologia assistiva pode fazer uso da semântica dos elementos para dar ênfase às informações da página. Exemplos detalhados de semântica aplicada à tecnologia serão explicados em seções de cabeçalhos e formulários.

5.3 TABELA ACESSÍVEL

O uso de tabelas se faz necessário quando precisamos apresentar dados tabulares. A principal recomendação para o uso de tabelas em uma página é que ela seja o mais simples possível. Lembre-se de que parte do público pode acessar a tabela utilizando um smartphone.

Quando utilizar uma tabela, lembre-se da semântica dos elementos:

- `<table>` - elemento para iniciar a tabela;
- `<caption>` - título e descrição da tabela;
- `<tr>` - linha da tabela;
- `<th>` - cabeçalho da linha/coluna;
- `<td>` - célula de dados.

Uma tabela simples ficaria da seguinte forma:

```
<table>
  <caption>Tabela de dados</caption>
  <tr>
    <th>Data</th>
    <th>Fonte</th>
    <th>Tamanho</th>
  </tr>
  <tr>
    <td>21 de abril</td>
    <td>Banco de dados</td>
    <td>125 Kb</td>
  </tr>
</table>
```

Ainda é possível adicionar mais informações, como detalhes de leitura da tabela no elemento `<caption>` e manipulá-lo pelo CSS:

```
<table>
  <caption>Tabela de dados
```

```
  <span>A leitura dos dados é feita de cima para baixo</span></ca
ption>
```

Este exemplo anterior é simples, mas para tabelas maiores e mais complexas é necessário considerar o uso de atributos para relacionar linhas e colunas:

```
<table>
  <caption>Horário de atendimento:</caption>
  <tr>
    <td></td>
    <th scope="col">Segunda-feira</th>
    <th scope="col">Terça-feira</th>
    <th scope="col">Quarta-feira</th>
    <th scope="col">Quinta-feira</th>
    <th scope="col">Sexta-feira</th>
  </tr>
  <tr>
    <th scope="row">09:00 - 11:00</th>
    <td>Aberto</td>
    <td>Aberto</td>
    <td>Aberto</td>
    <td>Fechado</td>
    <td>Aberto</td>
  </tr>
  <tr>
    <th scope="row">11:00 - 13:00</th>
    <td>Aberto</td>
    <td>Aberto</td>
    <td>Aberto</td>
    <td>Fechado</td>
    <td>Aberto</td>
  </tr>
</table>
```

O atributo `scope` serve para identificar onde começa a linha e a coluna da tabela. Os valores `col` (para coluna) e `row` (para linha) orientam a leitura.

Qual a barreira de acesso? Tabelas podem ser difíceis de interpretar quando não temos a orientação visual. Alguns recursos podem ajudar pessoas com deficiência (principalmente visual) a localizar a informação em dados tabulares.

5.4 IDIOMA DA PÁGINA

Muitas vezes relevamos ou não damos valor à marcação do idioma da página, porém ele tem um papel importante não só na internacionalização do conteúdo, mas também para a acessibilidade.

Definir o idioma da página é simples. Basta utilizar o atributo `lang` em qualquer elemento HTML, começando pelo elemento `html`:

```
<html lang="pt-br">
```

E a mudança do idioma pode ser feita a qualquer momento, em qualquer elemento:

```
Este texto está em português <span lang="en">and in english</span>
```

Os idiomas mais comuns são fáceis de lembrar: `en` para inglês, `fr` para francês, `de` para alemão, `es` para espanhol e `pt-br` para português do Brasil. Utilizar somente `pt` considerará o português de Portugal como padrão. Na página da IANA (https://www.iana.org/assignments/language-subtag-registry/language-subtag-registry) é possível encontrar todos os

valores para cada um dos idiomas utilizados no mundo e fora dele, porque até Klingon existe nessa lista.

Apesar de ser permitido pela documentação, nem sempre o atributo `lang` se comporta adequadamente no elemento `img`. Nesse caso, a recomendação é declarar o atributo em um elemento `container` onde está contida a imagem.

> **Qual a barreira de acesso?** Usuários de software leitores de tela que definem a identificação automática do idioma podem ter o sintetizador alterado automaticamente para o vocabulário de determinado idioma. Quando esse atributo não é definido, a leitura da página pode utilizar a configuração do leitor de tela e não necessariamente o vocabulário do idioma da página.

5.5 ESTRUTURA DE CABEÇALHOS

A navegação por cabeçalhos é uma da mais utilizadas por usuários de leitores de tela. Isso porque eles conseguem acessar todos os cabeçalhos apenas utilizando algumas teclas.

Para navegar, os usuários de leitores de tela utilizam a tecla `H` do teclado para navegar pelos cabeçalhos (e `shift + H` para voltar). Caso queira navegar por uma hierarquia específica, o usuário utiliza as teclas numéricas de `1` a `6` para navegar pelos cabeçalhos. Sendo assim, ao pressionar a tecla `2`, o usuário navega pelos cabeçalhos de nível 2 da página. Pressionando a tecla `3`, pelos cabeçalhos de nível 3, e assim por diante.

```
<h1>Jornal de notícias</h1>
  <h2>Esportes</h2>
    <h3>Grêmio segue para a próxima fase da Libertadores</h3>
    <h3>Seleção brasileira faz amistoso em São Paulo</h3>
  <h2>Variedades</h2>
    <h3>Lady Gaga lança novo disco</h3>
    <h3>Conheça a receita de bolo secreta da Palmirinha</h3>
```

QUAL A BARREIRA DE ACESSO? Sem uma estrutura adequada de cabeçalhos, usuários de tecnologia assistiva podem ter dificuldades em navegar pelas áreas do site.

5.6 LOCALIZAÇÃO

Possibilitar que o usuário se localize em uma página ou aplicação Web é muito importante para que ele não perca tanto o foco visual de navegação quanto em que área do site ele está navegando. Por isso, algumas técnicas que fazem parte desta publicação podem ser úteis para ajudar o usuário a se localizar em uma página:

- **Manter o foco do teclado visível**

Consiste em exibir de forma visual qual elemento está com o foco ativo por teclado. O detalhamento desta técnica será apresentado no próximo capítulo, em "navegação por teclado".

- **Criar títulos de páginas explicativos**

Além do que foi explicado neste capítulo sobre título da página, as informações disponíveis no elemento `title` devem descrever e ser explicativas sobre o que o usuário encontrará

naquela página.

- **Fornecer indicadores de localização (breadcrumbs)**

Breadcrumbs (migalhas de pão, em uma tradução literal, ou indicadores de localização) são aqueles caminhos encontrados nos sites para que o usuário consiga compreender a estrutura de navegação. Dessa forma fica mais fácil para que ele compreenda que a seção relacionada ao "corinthians" está dentro da seção "futebol", que por sua vez está dentro da seção de "esportes".

- **Fornecer um mapa do site ou lista de links e áreas importantes**

O mapa do site vem deixando de ser utilizado para dar espaço a buscas mais inteligentes e links para áreas importantes do site no rodapé das páginas, mas isso não significa que você não possa criar uma página com "mapa do site" se achar necessário.

> **Qual a barreira de acesso?** Quando o usuário não consegue se orientar em uma página, site ou aplicação Web isso se torna uma barreira que pode impedir não só a compreensão do conteúdo como a execução de tarefas na página ou aplicação.

5.7 SEQUÊNCIA DE LEITURA COM SIGNIFICADO

Quando existem diversos blocos de conteúdo em uma página,

eles devem ser posicionados de forma que o leitor consiga compreender a sequência de leitura adequada. Os elementos semânticos ajudam a compreender e dar peso semântico a conteúdo "principal" e "secundário" e com CSS é possível posicioná-los na página.

Acompanhe este exemplo de um post de esportes. A primeira é a notícia de destaque e a segunda, um bloco lateral de informação relacionada:

```
<div><h1>Santos vence o Corinthians na primeira rodada do campeon
ato </h1>
 <p>O time do técnico Jorge Sampaoli vence o rival Corinthians po
r dois a
   zero na última rodada do campeonato brasileiro.</p>
 <p>(continua a matéria)<p>
</div>
<div>A próxima rodada do campeonato acontece no sábado, dia 25, c
om o duelo entre São Paulo e Palmeiras</div>
```

Este exemplo pode parecer correto, pois apresenta uma notícia sobre Santos e Corinthians e depois São Paulo e Palmeiras. Pensando como uma publicação em uma página, bastaria um pouco de CSS para posicionar a segunda `<div>` na lateral esquerda da página. Visualmente não existem problemas, mas para quem está utilizando um recurso de tecnologia assistiva, essa informação pode parecer fazer parte da matéria anterior. Nesse caso, o ideal seria usar a marcação semântica para definir conteúdo principal e complementar (além de usar o CSS para posicionamento e formatação visual do bloco complementar):

```
<article><h1>Santos vence o Corinthians na primeira rodada do cam
peonato </h1>
 <p>O time do técnico Jorge Sampaoli vence o rival Corinthians po
r dois a
   zero na última rodada do campeonato brasileiro.</p>
 <p>(continua a matéria)<p>
```

```
</article>
<aside>A próxima rodada do campeonato acontece no sábado, dia 25,
 com o duelo entre São Paulo e Palmeiras</aside>
```

Dessa forma conseguimos dar peso semântico para a informação complementar, sem prejudicar a sequência de leitura. A customização pelo CSS pode permanecer a mesma e não tem diferença para quem enxerga, mas para quem usa leitores de tela essa estruturação do conteúdo é fundamental.

Esse foi um simples exemplo, mas é comum nos depararmos com páginas cheias de blocos de texto e contextos distintos. Blocos de texto fora de contexto espalhados no meio da página ou até mesmo publicidade obstrusiva podem dificultar a compreensão de pessoas com limitações cognitivas, que fazem uso de tecnologia assistiva ou que alteram propriedades de CSS para a melhor compreensão da informação.

Caso não seja possível preservar uma sequência de leitura compreensível a documentação do W3C orienta a proporcionar uma alternativa para o conteúdo na ordem adequada.

Qual a barreira de acesso? Pessoas com limitações cognitivas podem ter dificuldades em compreender uma informação se elas não forem apresentadas de forma sequencial ou se existir algo que obstrua parte da leitura. Se a customização das informações apresentadas for demasiadamente complexa, a leitura e assimilação do conteúdo da página pode ser comprometida.

Capítulo 6

ELIMINANDO BARREIRAS - NAVEGAÇÃO E INTERATIVIDADE

Um dos princípios da Web é a interatividade. Não somente para navegar entre links mas para executar ações como a compra de uma passagem aérea ou a solicitação de um documento por um órgão público.

Garantir a acessibilidade na navegação do usuário significa que ele deve não só conseguir operar a interface da Web como também compreender o seu comportamento e navegar de forma plena, independente de limitações técnicas ou físicas.

Muitas vezes a acessibilidade é relegada porque o usuário que tem barreiras de navegação não consegue nem chegar no campo adequado para fazer uma reclamação, de tão complexo que é o caminho para entrar em contato. Isso pode ser devido à falta de arquitetura da informação ou por barreiras de acessibilidade, que serão listadas ao longo deste capítulo.

Navegar e interagir de forma plena é um direito de todas as pessoas. A limitação ao acesso é uma barreira grave que pode causar consequências tanto para quem acessa (que não vai

conseguir operar aquela interface) quando para o proprietário do website, que vai perder público pelo simples fato de o usuário não conseguir navegar. Essa última afirmação já foi comprovada em uma pesquisa feita pelo *Movimento Web para Todos* sobre e-commerce, que mostra que, se o usuário encontra uma barreira no processo de escolher um produto, colocar no carrinho, fazer o cadastro ou efetuar o pagamento, ele pode não conseguir concluir a compra, e optar por ir para o site concorrente (https://mwpt.com.br/sites-de-e-commerce-nao-estao-preparados-para-receber-pessoas-com-deficiencia/).

Sendo assim, vamos conhecer como solucionar as principais barreiras de acesso para a navegação e interatividade.

6.1 NAVEGAÇÃO POR TECLADO

Talvez uma das principais regras da acessibilidade na Web seja: garantir que todo o conteúdo interativo esteja disponível via teclado. Isso significa que o que o usuário fizer com o mouse ele deve ser capaz de fazer utilizando a interface do teclado.

A navegação por teclado se dá por toques na tecla `tab` e `shift+tab` para navegar pelos elementos interativos e o acionamento desses elementos é feito utilizando a tecla `enter`. Quando o usuário utiliza um software leitor de tela ele costuma utilizar mais recursos do teclado, como as setas para ler o conteúdo e atalhos para elementos da página.

Foco visível: quando o usuário navega por teclado é comum que ele perca o foco ao passar por diversos links da página. Por isso, é importante que o foco seja visível, ou seja, permita uma

marcação quando passar o mouse ou fazer o foco por teclado em um item interativo na página. Por exemplo:

```
a{
background-color:blue;
color: white;
border: 2px solid blue;
}

a:hover, a:focus{
border: 2px solid yellow;
}
```

O foco visível pode ser customizado tanto na borda quanto no outline. No *box model*, o outline não ocupa espaço (ele vai ocupar espaço da margem do elemento), diferente da borda. Por isso é importante relembrar o conceito de *box model* do CSS:

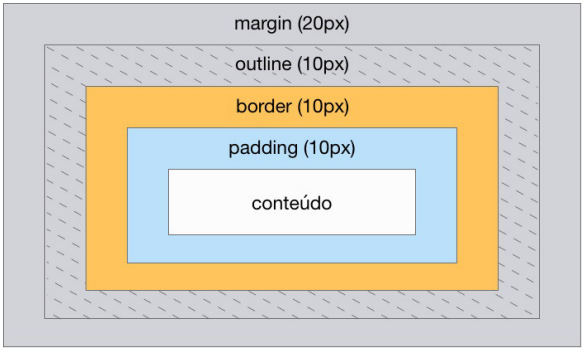


Figura 6.1: Imagem ilustrando o box model. De dentro para fora está o conteúdo, seguido do padding, border, outline e margin.

Embora não seja recomendado remover o outline (mas sim customizá-lo), se ele for removido do estilo dos links, deve ser fornecida outra alternativa de estilo para mostrar o foco visível, como borda e mudança de plano de fundo.

Essa customização de link por CSS acerta dois alvos com um

tiro só: permite que os links da página tenham uma borda amarela quando o usuário passar o mouse e quando fizer foco por teclado. Isso evita que o usuário se perca ao navegar na página e saiba em que link o foco atual está, além de permitir que a mesma função executada com o mouse seja executada também por teclado.

> **Qual a barreira de acesso?** Mesmo que sejam ações somente decorativas, o uso do recurso do foco visível ajuda os usuários a se localizarem na página, podendo ter certeza de que estão acionando o elemento correto. Sem esse recurso um usuário pode se perder e acionar links ou botões equivocadamente.

JavaScript obstrusivo: impede que o usuário consiga executar uma ação que por padrão é acessível por teclado. Esse recurso interfere principalmente quando se usam recursos somente para uso do mouse, como um `onClick` isolado. Isso impede que os usuários que navegam por teclado consigam acionar os elementos interativos.

No atual estado do padrão HTML já existem diversos elementos que possibilitam o uso de uma função de clique/acionamento por teclado como padrão, como botões. Mas se for extremamente necessário utilizar `onClick`, considere também o uso de `onKeyUp` ou `onKeyPress` para garantir que a função executada ao clicar seja a mesma quando acionada a tecla específica.

O mesmo serve para o JavaScript. Se você tem uma aplicação que executa uma função ao passar o mouse, o mesmo deve ocorrer

quando fizer o foco por teclado:

```
onmouseover="Event(this)"
onmouseout="Event(this)"
onfocus="Event(this)"
onblur="Event(this)"
```

Utilizar `onfocus` e `onblur` (em conjunto com `onmouseover` e `onmouseout`) possibilita que as ações que acontecem com o passar e retirar o mouse sejam também executadas quando o foco acontecer pelo teclado.

A navegação por teclado não beneficia somente quem navega por um teclado físico. Usuários de dispositivos móveis se beneficiam com essa técnica, já a navegação com tecnologia assistiva é feita com toques da esquerda para a direita para navegar por itens interativos de uma aplicação Web.

> **QUAL A BARREIRA DE ACESSO?** Quando uma ação exige do usuário o acesso exclusivo do clique ou passar o mouse, ela exclui o acesso de usuários que utilizam somente o teclado para navegar, como usuários de software leitores de tela e pessoas com mobilidade reduzida.

6.2 ACIONAMENTO POR GESTOS

Funcionalidades baseadas em toque devem considerar a possibilidade de o usuário conseguir executar a tarefa mesmo com limitações de toque e que ele possa cancelar uma ação que eventualmente tenha sido feita de forma incorreta.

Quando uma aplicação pede para o usuário ligar dois pontos na tela, por exemplo, deve existir uma possibilidade de se fazer isso sem a necessidade de seguir o caminho com o dedo. Pessoas com mobilidade reduzida podem ter dificuldade em seguir um caminho entre dois pontos e o toque no ponto inicial e no ponto final pode ser mais efetivo para que ele complete a ação.

Um outro bom exemplo é o uso de uma aplicação com um mapa, que permite *zoom in* e *zoom out* com o movimento de pinça em dispositivos móveis. Nesse caso, a recomendação é que, além do movimento de pinça, existam botões para que o usuário consiga fazer uso desse recurso, como no exemplo a seguir, do Google Maps.

Figura 6.2: Imagem mostrando botões de zoom in e zoom out no Google Maps

Qual a barreira de acesso? Pessoas que não conseguem fazer movimento de pinça por limitações físicas ou pelo uso de um dispositivo sem interface *touch* podem não conseguir usar o mapa.

6.3 ATALHOS DE TECLADO E ACIONAMENTO POR VOZ

Com o avanço de tecnologia de reconhecimento e comando de voz é importante considerar que certos atalhos utilizados para teclado podem trazer barreiras para esse tipo de acionamento.

Imagine uma aplicação que utiliza atalhos para teclado. Na aplicação a tecla "M" desliga o áudio, a tecla "N" avança para a próxima mensagem e a tecla "P" apaga a mensagem.

Agora considere que uma pessoa utilizando um smartphone com seu assistente pessoal habilitado fale algo parecido com essa frase:

Pedro! Você viu a **Ammy** por ai?

Os sons das letras **pê** e **eme** na frase podem acionar os atalhos de teclado. No nosso exemplo, ele poderia apagar uma mensagem ou passar para a próxima mensagem.

A utilização de atalhos deve ser feita com cuidado ou com combinação de teclas para evitar esse tipo de situação. Caso utilize atalhos de teclado, deve ser possível para o usuário desligar,

remapear ou ativá-lo apenas com foco.

Lembre-se de que todo o acionamento por voz também deve ter uma interface pela qual o usuário consiga utilizar a partir de uma interface, pois o usuário pode não conseguir falar ou ter um microfone disponível para acionar a aplicação a todo momento.

> **Qual a barreira de acesso?** Pessoas que utilizam sistemas de reconhecimento de voz podem acionar acidentalmente atalhos de teclados por assistentes pessoais.

6.4 SALTAR CONTEÚDO REPETIDO

Imagine um sistema de e-commerce com dezenas de links para navegação. Agora pense na dificuldade de uma pessoa que navega por teclado ao acessar uma informação no meio da página.

Para minimizar esse problema você pode utilizar uma solução simples e muito eficiente: Colocar um link no topo do site para o conteúdo.

Utilize um elemento de âncora que vai direcionar o foco para o `ID` de um elemento da página, por exemplo:

```
<a href="#conteudo">Saltar conteúdo</a>
…
<section id="conteudo">
```

Ao utilizar essa técnica como o primeiro ou segundo link da página, o usuário pode acioná-lo e ser direcionado diretamente para o conteúdo, sem a necessidade de passar por todos os links da

página.

Essa técnica guarda um dilema: manter ou não esse link escondido?

Isso é um dilema porque algumas pessoas defendem que usuários de software leitores de tela não precisam ver esse link e é uma informação visual indispensável. O problema é que o deixar invisível faz com que pessoas com limitações motoras possam não perceber esse link.

Uma forma de resolver isso é manter esse link escondido e quando o usuário fizer o foco por teclado você exibi-lo na página. Se o usuário continuar a navegar por teclado, esse link volta a ficar escondido.

O site *The Web Standards Project* (https://www.webstandards.org/) é um bom exemplo de uso dessa solução:

Na primeira imagem está o cabeçalho do site, sem botões de salto para conteúdo.

WEB STANDARDS PROJECT
The Web Standards Project is a grassroots coalition fighting for standards which ensure simple, affordable access to

Figura 6.3: Captura de tela do cabeçalho do site "The Web Standards Project"

Quando o usuário começa a navegar pela tecla "tab" ele chega ao link para "saltar para o conteúdo" no terceiro toque. Ao chegar

lá, uma barra aparece no topo do site (e desaparece quando o foco é retirado):

Skip to content
WEB STANDARDS PROJECT
The Web Standards Project is a grassroots coalition fighting for standards which ensure simple, affordable access to

Figura 6.4: Captura de tela do cabeçalho do site "The Web Standards Project" exibindo o texto "Skip to Content" no topo da página

Qual a barreira de acesso? Pessoas com limitações motoras e usuários de software leitores de tela podem ter dificuldade em navegar em páginas com muitos links. Isso ajuda o usuário a saltar conteúdo interativo desnecessário.

6.5 ORDEM DE FOCO

Quando o usuário navega por teclado, ele espera que o fluxo de navegação seja previsível. Normalmente, o usuário começa a navegação pelo topo da página e vai descendo, passando por eventuais menus superiores e laterais até chegar ao conteúdo.

Esse tipo de navegação previsível ajuda o usuário a não só saber onde ele está na página como o caminho que ele deve seguir, se continua navegando para o próximo link ou não.

Por isso, é fundamental pensar na ordem de navegação pelo

teclado. Normalmente publicamos o código conforme sua ordem de leitura e essa navegação flui normalmente (de cima para baixo e da esquerda para a direita). Mas, caso isso seja impossível, pode-se usar o atributo `tabindex` com o valor da navegação. Os valores são numéricos e sempre seguem a sequência do menor para o maior.

Alguns estudos mostram que utilizar valores maiores que "0" no `tabindex` pode bagunçar a ordem de navegação por teclado e complicar a navegação em páginas dinâmicas. Um artigo bem interessante sobre o assunto detalha essa técnica no link https://adrianroselli.com/2014/11/dont-use-tabindex-greater-than-0.html.

> **Qual a barreira de acesso?** Aplicações nas quais não se consegue navegar por teclado de forma sequencial pode causar dificuldades para pessoas que não enxergam ou com mobilidade reduzida. Esse recurso permite que o usuário não seja surpreendido por uma navegação confusa ou diferente dos padrões com que ele está acostumado.

6.6 FORMULÁRIOS

Este talvez seja um dos elementos interativos mais importantes da Web, já que com ele obtemos informações voluntárias dos usuários. Um formulário tem que ser planejado adequadamente pois uma experiência ruim do usuário pode causar o abandono do preenchimento de uma simples página de contato ou de um

sistema de pagamento de um e-commerce.

Relacionando campos e rótulos

A primeira regra do formulário é que os campos devem estar relacionados com seus rótulos. A marcação para isso é muito simples:

```
<label for="nome">Nome</label><input type="text" id="nome" value=
"nome">
```

Fazer a relação entre o campo e o rótulo permite que o usuário saiba exatamente que tipo de informação encontrar na página. A sintaxe dessa relação pode ser feita de outra forma:

```
<label for="nome">Nome<input type="text" id="nome" value="nome"><
label>
```

Ou mesmo:

```
<label>Nome<input type="text" id="nome" value="nome"></label>
```

Sempre que puder, faça a relação com o atributo `for` do elemento `<label>` . Isso dá muito mais significado à relação entre o campo e o formulário.

O mesmo se aplica para `radiobox` e `checkbox` , mas nesse caso o texto deve vir depois do campo:

```
<input type="radio" name="sexo" value="masculino" id="masc">
<label for="masc">Masculino</label>
```

A vantagem do uso dessa técnica é que o usuário não precisa mais clicar somente no campo para selecioná-lo. Ao clicar no texto o campo também é selecionado.

Caso não seja possível utilizar o elemento `<label>` por algum

motivo, utilize o atributo `aria-label` ou `title` para descrever o campo de formulário:

```
Telefone:<input type="tel" name="ddd" aria-label="Preencha o códi
go DDD"> <input type="tel" name="tel" aria-label="Preencha com o
seu número de telefone>
```

Recomenda-se não utilizar os dois atributos ao mesmo tempo.

> **QUAL A BARREIRA DE ACESSO?** Software leitores de tela relacionam o texto com o campo de formulário. Sem essa informação, o usuário pode não conseguir identificar o campo e preenchê-lo equivocadamente. Para campos `radiobox` e `checkbox` existe um ganho de usabilidade, já que a área de acionamento aumenta e não é mais necessário clicar na "bolinha" ou "quadradinho" do campo.

Você também pode fazer uso dos `input type` do HTML5 para trazer mais significado para os campos de formulário. Por exemplo, ao utilizar o `input type="tel"` em um campo de telefone, o dispositivo pode mudar automaticamente para o teclado numérico, afinal, esse campo só requer números.

O mesmo vale para o `input type="email"` , que já exibe o símbolo de `@` no teclado, e o `input type="url"` , que exibe um botão `.com` para facilitar o preenchimento na maioria dos dispositivos móveis.

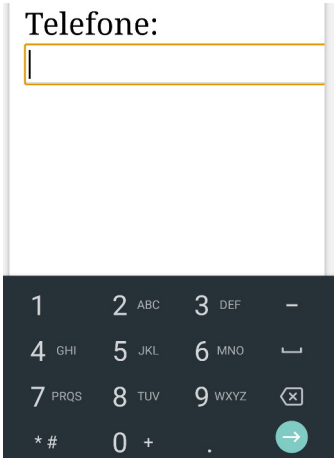


Figura 6.5: Exibição do teclado numérico quando o foco do teclado faz foco no campo de telefone

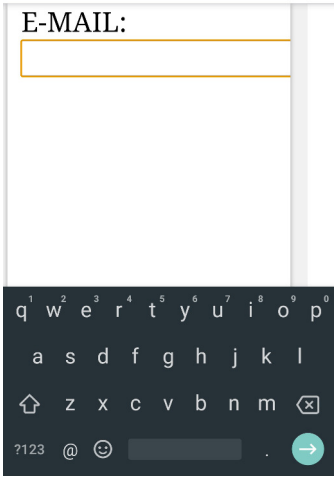


Figura 6.6: Exibição do teclado com símbolo de arroba quando o foco do teclado faz foco no campo de email

A lista completa dos `input type` do HTML5 está disponível no capítulo anterior deste livro.

Qual a barreira de acesso? Utilizar os elementos semânticos do HTML5 ajuda a identificar os campos de formulário e deixar seu preenchimento mais fácil e intuitivo. Isso é importante principalmente para pessoas com limitações cognitivas, que podem ser beneficiadas com um teclado mais simples para o preenchimento de dados específicos.

Agrupando campos de formulário

Você também pode agrupar campos de formulário para facilitar o preenchimento do usuário.

Imagine a situação: seu usuário precisa preencher um formulário com o endereço pessoal e comercial. O código do formulário é o seguinte:

```
Endereço Comercial
<label for="end-com">Endereço:</label>
<input type="text" id="end-com" name="end-com>
<label for="num-com">Número:</label>
<input type="text" id="num-com" name="num-com>
<label for="bai-com">Bairro:</label>
<input type="text" id="bai-com" name="bai-com>
<label for="cep-com">CEP:</label>
<input type="text" id="cep-com" name="cep-com>

Endereço Residencial
<label for="end-res">Endereço:</label>
<input type="text" id="end-res" name="end-res>
<label for="num-res">Número:</label>
<input type="text" id="num-res" name="num-res>
<label for="bai-res">Bairro:</label>
<input type="text" id="bai-res" name="bai-res>
<label for="cep-com">CEP:</label>
<input type="text" id="cep-res" name="cep-res>
```

Porém, quando um usuário de software leitor de tela acessar a página, a informação será exibida da seguinte forma quando o usuário navegar por teclado entre os campos:
Campo: Endereço -> Campo: Número -> Campo: Bairro -> Campo: CEP -> Campo: Endereço -> Campo: Número -> Campo: Bairro -> Campo: CEP ->

Dessa forma não existe diferenciação entre os tipos de dados requeridos e o usuário terá que navegar nas informações que existem antes e depois para compreender o que significa.

Para solucionar essa barreira, basta utilizar alguns elementos específicos para agrupamento de campos de formulário. Os elementos `<fieldset>` e `<legend>` permitem que os grupos de formulário sejam identificados e marcados. O elemento `<fieldset>` permite o agrupamento dos campos de formulário enquanto o `<legend>` permite dar um nome a esse grupo.

Este exemplo de código tem a marcação adequada:

```
<fieldset>`
<legend>Endereço Comercial</legend>
<label for="end-com">Endereço:</label>
<input type="text" id="end-com" name="end-com>
<label for="num-com">Número:</label>
<input type="text" id="num-com" name="num-com>
<label for="bai-com">Bairro:</label>
<input type="text" id="bai-com" name="bai-com>
<label for="cep-com">CEP:</label>
<input type="text" id="cep-com" name="cep-com>
</fieldset>

<fieldset>
<legend>Endereço Residencial</legend>
<label for="end-res">Endereço:</label>
<input type="text" id="end-res" name="end-res>
<label for="num-res">Número:</label>
<input type="text" id="num-res" name="num-res>
```

```
<label for="bai-res">Bairro:</label>
<input type="text" id="bai-res" name="bai-res>
<label for="cep-com">CEP:</label>
<input type="text" id="cep-res" name="cep-res>
</fieldset>
```

Quando um software leitor de tela acessar esse formulário ele vai ler o conteúdo interativo desta forma:

Grupo: Endereço Comercial -> Campo: Endereço -> Campo: Número -> Campo: Bairro -> Campo: CEP -> Grupo: Endereço Residencial -> Campo: Endereço -> Campo: Número -> Campo: Bairro -> Campo: CEP ->

O código anterior resulta na seguinte renderização em um navegador:

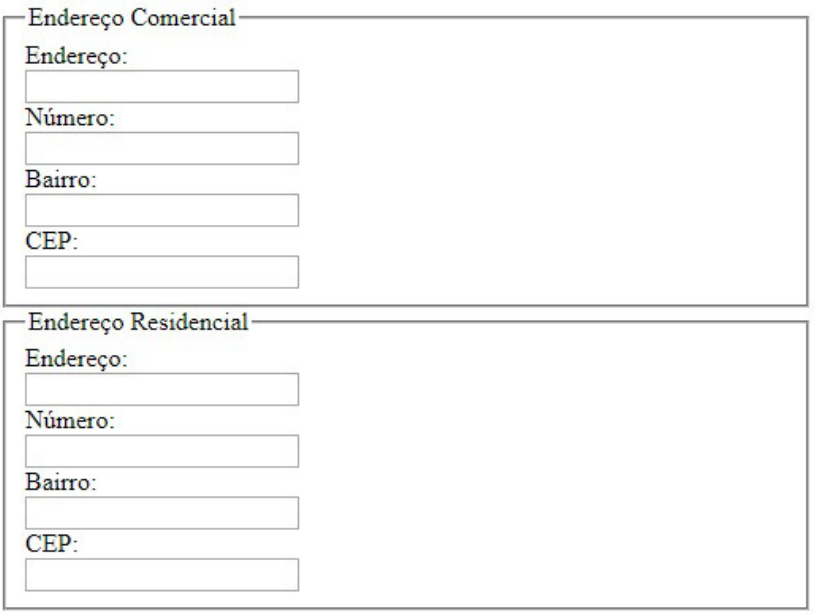


Figura 6.7: Renderização do exemplo de código dos campos de formulário. O exemplo mostra a divisão dos blocos do formulário quando utilizados os elementos fieldset

A exibição no navegador mostra claramente a separação dos blocos do campo de formulário, bem como as informações de quais dados são necessários para cada um deles.

Esse tipo de técnica facilita o preenchimento do formulário para qualquer usuário, seja ele um usuário de software leitor de tela ou não.

> **Qual a barreira de acesso?** Formulários não agrupados e identificados podem causar dificuldades para o usuário, já que a informação requerida no campo não fica clara quando os campos não estão relacionados adequadamente.

Botões acessíveis

Pode parecer exagero considerar um tópico exclusivamente para botões acessíveis, mas ele é necessário. Apesar de existirem elementos específicos para uso de botões, é muito comum a utilização de outros elementos para exercer a função de botões. Mas isso pode trazer mais problemas do que soluções.

Não é raro encontrarmos elementos e *containers* genéricos que se comportam como botões. Como este exemplo:

```
<span onClick="event()">Submeter proposta</span>
```

Considere que este botão está rodando um script que, ao ser clicado, envia os dados do formulário, e que ele está estilizado como um botão:

Figura 6.8: Botão Submeter Proposta

Existem diversos problemas neste botão. O mais grave é que ele vai se comportar como botão somente quando o usuário clicar com o mouse. Não dá para chegar a este campo de formulário utilizando a navegação por teclado. Para que a navegação por teclado funcione, é necessário acrescentar alguns atributos tanto de CSS quanto de JavaScript para isso:

```
<span role="button"
tabindex="0"
onClick="event()"
onKeyPress="event()"
onFocus="event()"
onBlur="event()"
onMouseOver="event()"
onMouseOut="event()"
>Submeter proposta</span>
```

A descrição das funções de cada botão é a seguinte:

O atributo `role="button"` muda a semântica do elemento. O *container* genérico de texto `<span>` passa a se comportar semanticamente como um botão.

O atributo `tabindex` possibilita que um elemento, que não é interativo por padrão, tenha o foco disponível por teclado.

O atributo `onKeyPress` possibilita que esse botão seja executado quando uma tecla seja pressionada. Neste caso, o ideal é que sejam as teclas `space` e `enter`.

`OnFocus` muda o comportamento quando o usuário fizer o foco por teclado. Deve ser o mesmo comportamento de `OnMouseOver` quando o usuário passar o mouse.

`OnBlur` muda o comportamento quando o usuário tirar o foco do teclado. Deve ser o mesmo comportamento de `OnMouseOut` quando o usuário tirar o mouse do elemento.

Esses atributos, além de serem aplicados no elemento, ainda precisam ter seu comportamento configurado no JavaScript.

Também é necessário adicionar no CSS o seguinte código para que visualmente o botão se comporte adequadamente quando se passar o mouse ou fazer o foco por teclado:

```
span:hover,
span: focus{
background-color:gray;
}
```

Os elementos `<button>`, `<input type="image">` e `<input type="submit">` já possuem todos esses recursos interativos por padrão, sendo selecionáveis pela navegação com mouse ou por teclado. Elementos de âncora, como `<a>`, também costumam ser utilizados, mas o seu atributo `href` algumas vezes pode causar problemas para acessibilidade.

Qual a barreira de acesso? Pessoas que navegam por teclado precisam conseguir executar as tarefas da mesma forma que uma pessoa que utiliza o mouse deve fazer. Por isso, considerar a navegação por teclado garante que pessoas com limitações motoras consigam navegar sem o uso do mouse. Para usuários de leitores de tela o impacto é maior, pois se os elementos não são semanticamente marcados como botões, é necessário não só garantir que seu acesso por teclado esteja disponível como a marcação de que aquele elemento vai se comportar como um botão.

Para botões mais complexos existe a possibilidade de uso de atributos de WAI-ARIA para estender a descrição deles. Imagine que você tem uma aplicação que liga ou desliga o som de uma página. Para um usuário de leitores de tela essa informação pode não ser clara. Por isso o uso de atributos de WAI-ARIA é necessário.

Quando o usuário chegar em um botão para habilitar o áudio, ele pode encontrar o seguinte código:

```
<div role="button" tabindex="0" aria-pressed="true">Desligar som<
div>`
```

Além do script para habilitar e desligar o som, é necessário garantir que o botão esteja acessível via teclado e também que o valor de `aria-pressed` mude para `false` . Essa informação é importante para que o usuário de leitores de tela saiba que o estado daquele botão mudou depois de pressionado

Qual a barreira de acesso? Usuários de leitores de tela nem sempre são informados de mudanças que acontecem na página durante a interação com botões. Utilizar atributos de WAI-ARIA ajuda a estender essa descrição, informando ao usuário a mudança de status do botão.

Instruções e mensagens de erro

Utilizar uma aplicação Web não é simples para todas as pessoas. Algumas podem ter dificuldades por não compreender como executar as ações exigidas em uma página ou podem ter limitações relacionadas à falta de acessibilidade da aplicação.

Por esse motivo, é importante considerar que as instruções e mensagens de erro sejam claras para que as pessoas possam compreender as informações e utilizar a aplicação plenamente.

Além de relacionar os rótulos com os campos de formulário e agrupar os campos conforme os dados solicitados (como visto previamente neste capítulo), devemos informar ao usuário o que esperar daquela aplicação. Vamos dar um exemplo de um formulário para cadastro em um site.

Campos obrigatórios devem estar claramente marcados para o usuário. Para tanto, utilize elementos adicionais aos rótulos de campos obrigatórios. Você pode utilizar textos dentro do elemento `<label>` informando ao usuário que aquele campo é obrigatório, e utilizar o elemento `<span>` para alterar a exibição se necessário.

```
<label for="nome">Nome <span>(obrigatório)</span></label>
```

```
<input type="text" id="nome" name="nome" aria-required="true">
```

Você também pode utilizar o atributo `aria-required`, que informa ao usuário de tecnologia assistiva que aquele campo é obrigatório.

Mas dá para deixar a aplicação mais inteligente e transmitindo mais informações ao usuário, principalmente aos usuários de software leitores de tela.

Quando o campo de formulário exigir informações mais específicas, utilize o atributo `aria-describedby` para relacionar essa informação com o rótulo, como no exemplo a seguir:

```
<label for="nome">Data de nascimento <span>(obrigatório)</span></
label>
<input type="text" id="nome" name="nome" aria-describedby="comple
mentar">`
```

E em algum ponto da página, seja ele antes ou depois do campo de formulário, complemente a informação com o seguinte parágrafo:

```
<p id="complementar">Utilize o formato dia/mês/ano com dois dígit
os para cada.</p>
```

Neste outro exemplo, considere um elemento `<select>` de países que muda automaticamente o próximo campo exibindo as cidades. É possível utilizar essa técnica com `aria-describedby` para informar ao usuário que o próximo campo será atualizado.

```
<select id="paises" aria-describedby="aviso">
<option id="01">Argentina</option>
<option id="02">Brasil</option>
<option id="03">Canadá</option>
...
</select>
<p id="aviso">Ao selecionar um país o próximo campo será atualiza
```

```
do automaticamente carregando os estados</p>
```

A vantagem dessa técnica é proporcionar ao usuário a informação necessária antes que ele mude o foco para o próximo campo. Utilizando esse atributo o usuário de tecnologia assistiva é informado quando o foco do teclado atingir o campo marcado.

Quando trabalhar as mensagens de erro em um site, considere as seguintes questões:

- O usuário deve saber que existem erros;
- O usuário deve conseguir identificar e atingir os campos com erro;
- O usuário deve conseguir corrigir os campos.

Nesse contexto, uma técnica que pode ser útil é colocar as mensagens de erro com links para cada um dos campos com erro, por exemplo:

```
<div id="erro">Erros encontrados no formulário:
<ul>
<li><a href="#nome">Preencha o campo Nome</a></li>
<li><a href="#data">Preencha a data com o formato adequado</a></l
>
</ul>
</div>
```

Considerando que os campos de formulário têm seus atributos `id` descritos da forma correta, essa técnica vai direcionar cada link para o campo específico para que o usuário consiga atingir os campos e corrigir as informações neles.

É importante também que, quando o formulário identificar um erro que não pode seguir adiante, o foco vá para o elemento *container* das mensagens de erro. Neste exemplo, o elemento `<div>` com o `id="erro"`.

Lembre-se de informar ao usuário os tipos e formatos de dados que são necessários para o preenchimento do formulário.

> **QUAL A BARREIRA DE ACESSO?** Nem todos os usuários são experientes ou têm as referências visuais para compreender aplicações complexas e campos de formulários. Por esse motivo, é necessário que tanto as informações de preenchimento quanto as informações relacionadas às mensagens de erro sejam simples e de fácil execução.

6.7 ACIONAMENTO E CANCELAMENTO DE BOTÃO

Outro fator importante em aplicações de toque é a possibilidade de o usuário cancelar uma ação disparada acidentalmente. É muito comum que um usuário que pressione equivocadamente um botão tenha a oportunidade de cancelar esse toque. Imagine um campo de formulário todo preenchido, e quando o usuário termina de preencher ele pressiona o botão "Limpar" em vez do botão "Enviar"?

Os elementos interativos (como links e botões) só são acionados quando o usuário “solta” o botão do mouse ou o toque na tela. Esse comportamento padrão ajuda pessoas que possam ter clicado equivocadamente no botão e podem arrastar o cursor para fora da área do botão ou link para evitar o acionamento.

Por esse motivo é recomendado não utilizar o evento `down` . Esse evento pode ser conhecido de diversas formas, como

`mousedown`. Utilizar `mousedown` faz com que o acionamento aconteça no toque, e não quando o usuário soltar o botão. Prefira utilizar `mouseup`, que somente aciona o elemento interativo ao soltar o mouse.

O uso dessa técnica permite que o usuário que tocou equivocadamente em um link possa arrastar o dedo para fora da área do elemento interativo, evitando que ele seja acionado.

Caso o uso de `mousedown` seja necessário, deve existir uma forma de o usuário cancelar a operação.

Os elementos interativos do HTML (como âncoras e botões) são acionados pelo evento "up" por padrão. Esse recurso funciona tanto para toque como para o uso de mouse em computadores.

> **Qual a barreira de acesso?** Pessoas com limitações motoras podem tocar equivocadamente em um botão ou link e fazer com que ações inesperadas, como recarregar a página ou submeter um formulário, sejam acionadas. Permitir o acionamento somente ao soltar o elemento interativo faz com que o usuário evite o acionamento equivocado.

6.8 PROPÓSITO DO LINK

O usuário deve compreender o que encontrará ao clicar em um determinado link pelo seu próprio texto. É muito comum em sites de notícias encontrar os famosos "saiba mais" ou "leia mais" e que fazem com que o usuário tenha que identificar os textos próximos

para entender para onde aquele link vai direcioná-lo.

Por exemplo, em vez de colocar um link de “saiba mais”, esse link pode ser o título da notícia que está sendo apresentada ou um trecho do texto da chamada. Tudo depende da forma como esse conteúdo é apresentado.

Você também pode aninhar elementos para tratar toda a área de uma chamada de notícia como link, evitando o "saiba mais":

```
<a href="link.html">
<h2>Santos elimina o Corinthians da Copa do Brasil<h2>
<p>O time da Vila Belmiro virou o jogo contra o adversário e está
 nas finais do campeonato</p>
</a>
```

Links no meio do texto podem ser utilizados, desde que representem aquilo que efetivamente apresentarão para o usuário:

```
<p>O <a href="relatorio.html">Relatório de Vendas</a> foi produzi
do pela equipe de sistemas</p>
```

O hiperlink tem um papel fundamental na experiência do usuário. Possibilitar o propósito do link evita que o usuário se perca na página. É muito mais intuitivo para o usuário quando ele encontra um link com o texto “Acesse a agenda de cursos” do que apenas “Saiba mais”.

Você também pode indicar no link quando ele abre uma nova janela:

```
<p>O <a href="relatorio.html">Relatório de Vendas (abre em uma no
va janela)</a> foi produzido pela equipe de sistemas</p>
```

Ou:

```
<p>O <a href="relatorio.html">Relatório de Vendas <img src="icone
.png" alt="abre em uma nova janela"></a> foi produzido pela equip
```

```
e de sistemas</p>
```

Ou para informações de tamanho de arquivo para download:

```
<p>O <a href="relatorio.pdf">Relatório de Vendas <img src="icone-
pdf.png" alt="Arquivo pdf">(1.2 Mb)</a> foi produzido pela equipe
 de sistemas</p>
```

```
<p>O <a href="relatorio.pdf">Relatório de Vendas (arquivo em PDF
- 1.2 Mb)</a> foi produzido pela equipe de sistemas</p>
```

Essa técnica é importante por fornecer previsibilidade e também não faz com que a pessoa baixe um arquivo inadvertidamente.

> **Qual a barreira de acesso?** Principalmente pessoas com deficiência visual e usuários de software leitores de tela podem ter dificuldade de compreender um link fora de seu contexto, já que sua navegação costuma ser primordialmente por teclado e pelos elementos interativos da página.

6.9 TEMPO SUFICIENTE

Usuários devem ter tempo suficiente para interagir ou controlar aplicações Web. Páginas que não permitem esse controle podem dificultar o consumo por pessoas com deficiência, mas também podem atrapalhar a leitura de documentos, principalmente em smartphones.

Isso significa que o usuário deve ter tempo para fechar uma janela que abre automaticamente (sem a intervenção do usuário) antes que ela seja aberta, ou que ele consiga ler ou interromper

uma atualização automática da página.

A atualização automática pode ser uma barreira para todos os usuários. Qualquer pessoa que esteja lendo um texto longo ou preenchendo um extenso formulário pode ter problemas se a página for atualizada automaticamente, já que essa atualização mandaria o foco da leitura para o topo da página e poderia apagar tudo o que o usuário já preencheu no formulário.

Qual a barreira de acesso? Permitir que o usuário tenha tempo suficiente para executar tarefas (como ler ou concluir ações em uma aplicação) garante que os usuários não se percam devido ações repentinas ocorridas na página. Isso pode ser um transtorno principalmente para pessoas com deficiência visual, intelectual, motora e também para pessoas que tem um ritmo de leitura mais lento.

6.10 TIMEOUTS

O usuário deve ser avisado quando alguma aplicação ou recurso expirar devido à inatividade do usuário. Isso é muito importante porque o usuário pode perder dados caso a aplicação encerre sem que ele seja alertado.

Recarregamento automático também deve ser evitado, já que o usuário pode não conseguir ler ou operar a aplicação em tempo hábil antes de a página ser recarregada.

Qual a barreira de acesso? Pessoas com limitações motoras ou com tecnologia assistiva podem necessitar de mais tempo para cumprir tarefas da aplicação. A informação do tempo que ela tem disponível para executar a tarefa é fundamental para evitar que ela perca tudo o que está fazendo por timeout.

6.11 ATUALIZAÇÃO AUTOMÁTICA DE ÁREAS DO SITE

Antigamente para atualizar algo em uma página ou aplicação era necessário utilizar a atualização automática (`refresh`) em toda a página, mesmo que a atualização ocorresse em apenas uma área da página. Desde a difusão de Ajax (que possibilita atualização automática de algumas áreas do site) é possível atualizar somente algumas áreas específicas. Mas como fica a acessibilidade desse conteúdo dinâmico?

Por padrão, nem toda tecnologia assistiva lê atualizações em partes da página. Para que isso aconteça, é necessário utilizar alguns atributos de WAI-ARIA.

Primeiro, é necessário definir a área que sofrerá atualização. Definimos com o atributo `role` a área que será atualizada. Em seguida, é necessário definir se a área que vai ser atualizada será anunciada por tecnologia assistiva, utilizando o atributo `aria-live` .

Existem diversos tipos de regiões que podem ser utilizadas para anunciar atualizações. São elas:

- `role:log`
- `role:status`
- `role:alert`
- `role:progressbar`
- `role:marquee`
- `role:timer`

Os atributos a seguir servem para orientar tecnologia assistiva para ler ou não conteúdo que está sendo atualizado:

- `aria-live: polite` : inicia a leitura do conteúdo ao término da anterior;
- `aria-live: assertive` : interrompe a leitura atual e inicia a seguinte;
- `aria-live: off` : não lê o conteúdo atualizado.

Um exemplo simples de um elemento utilizando `aria-live` ficaria assim:

```
<aside role="alert" id="update" aria-live="polite">
 … conteúdo atualilzado...
</aside>
```

Qual a barreira de acesso? Aplicações dinâmicas e complexas muitas vezes criam barreiras para pessoas com deficiência, principalmente para usuários de software leitores de tela. Utilizar esse recurso permite expor para o usuário o conteúdo que será atualizado e que será lido ou não pelo leitor de tela.

6.12 ATIVAÇÃO POR MOVIMENTO

Essa diretriz surgiu com o objetivo de tornar acessível controles e comandos que possam ser executados com movimentos do usuário, seja ele movimentando o dispositivo ou movimentando seu corpo.

A questão é que essa diretriz não é impeditiva, e sim complementar ao recurso de ativação do movimento.

Quando desenvolver uma aplicação que será ativada por movimento (por exemplo, uma aplicação que faz login com o balançar do celular ou que ativa a câmera ao passar a mão sobre ela), deve existir uma forma de ativar o mesmo recurso pela interface do usuário. Isso significa que, além do próprio recurso de movimento, deve existir algo (seja um botão ou um link) para ativar esse recurso pela interface da aplicação.

Qual a barreira de acesso? Pessoas com mobilidade reduzida podem não conseguir acionar recursos ativados por movimento devido a complexidade de movimentos exigidos pela aplicação. Caso não consigam, elas podem fazer isso por controles da própria aplicação.

Capítulo 7

ELIMINANDO BARREIRAS - DESIGN

Quando tratamos a acessibilidade no design de uma aplicação levamos em consideração técnicas que vão muito além do código. Tudo isso para garantir que a percepção do usuário seja adequada e que barreiras sensoriais não interfiram na forma como se consome conteúdo na página. O design deve levar em conta a experiência do usuário, suas percepções de formas, cores e posições de elementos em uma página.

7.1 DESIGN RESPONSIVO E ACESSIBILIDADE

Pode parecer não haver relação entre o design responsivo e acessibilidade de uma aplicação, mas basta acessar uma página que não funciona adequadamente em um dispositivo móvel para compreender como esses dois temas têm relação direta entre si. Nesse aspecto, poderíamos elencar pelo menos três situações onde o design responsivo beneficia a acessibilidade digital.

Primeiro porque permite uma melhor navegação na página. Evitar barras de rolagem lateral dá maior conforto para a navegação no dispositivo móvel. Por exemplo, quando a página não possui responsividade e é exibida por completo, o usuário tem

que usar a barra de rolagem inferior para ler o texto ou utilizar dois dedos na tela mover a área de visualização. Nesse caso vale relembrar que nem todas as pessoas conseguem fazer esse movimento com os dois dedos para navegar pelas páginas.

Segundo porque, quando o design responsivo não é considerado, a leitura fica prejudicada. Páginas e aplicações que não se adaptam ao tamanho do dispositivo necessitam de intervenção do usuário para zoom e navegação e esta situação também envolve gestos e movimentos que nem todos os usuários podem executar.

E terceiro, e talvez mais importante, o design responsivo facilita o toque em links muito pequenos. O risco de o usuário pressionar um link ou botão inadequadamente é muito grande quando ele precisa dar zoom na tela para enxergar um determinado conteúdo.

O exemplo a seguir mostra a exibição de um site com design responsivo e outro sem. Mesmo sendo um site simples, a diferença no tamanho das fontes já interfere na leitura e no acesso a elementos interativos.

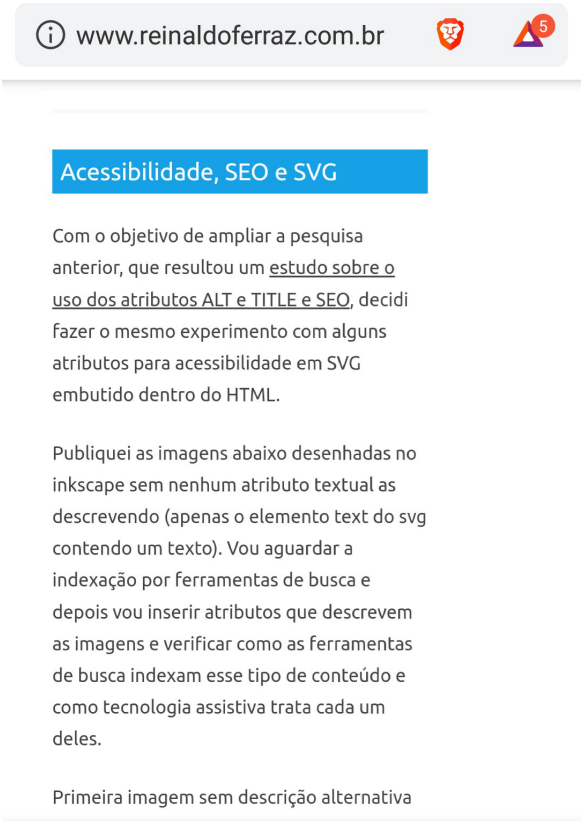

www.reinaldoferraz.com.br
Acessibilidade, SEO e SVG
Com o objetivo de ampliar a pesquisa anterior, que resultou um estudo sobre o uso dos atributos ALT e TITLE e SEO, decidi fazer o mesmo experimento com alguns atributos para acessibilidade em SVG embutido dentro do HTML.
Publiquei as imagens abaixo desenhadas no inkscape sem nenhum atributo textual as descrevendo (apenas o elemento text do svg contendo um texto). Vou aguardar a indexação por ferramentas de busca e depois vou inserir atributos que descrevem as imagens e verificar como as ferramentas de busca indexam esse tipo de conteúdo e como tecnologia assistiva trata cada um deles.
Primeira imagem sem descrição alternativa

Figura 7.1: Imagem de um site responsivo

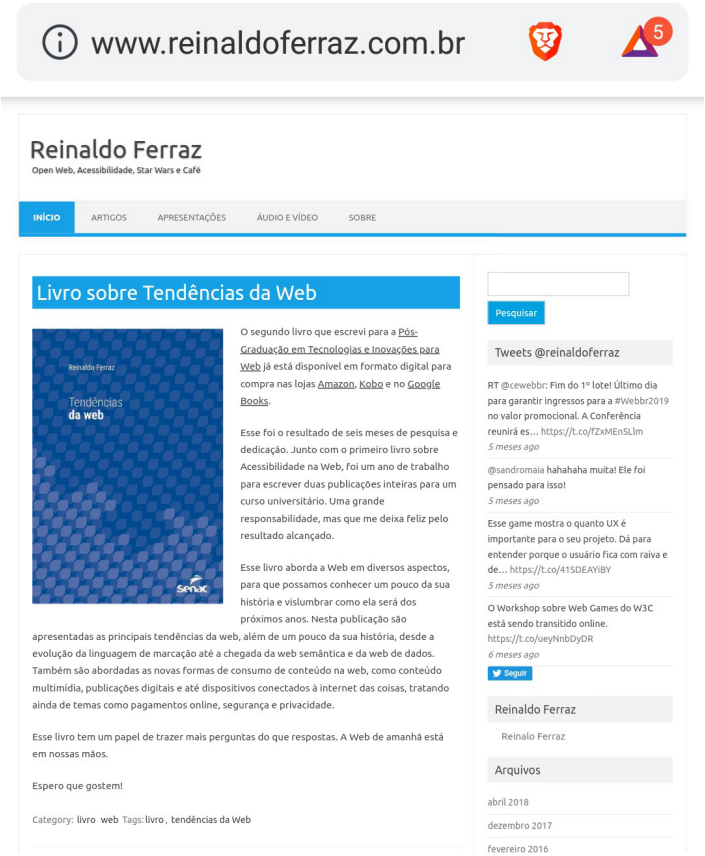

www.reinaldoferraz.com.br
Reinaldo Ferraz
Open Web, Acessibilidade, Star Wars e Café
INÍCIO
ARTIGOS
APRESENTAÇÕES
ÁUDIO E VÍDEO
SOBRE
Livro sobre Tendências da Web
Reinaldo Ferraz
Tendências da web
Senac
O segundo livro que escrevi para a Pós-Graduação em Tecnologias e Inovações para Web já está disponível em formato digital para compra nas lojas Amazon, Kobo e no Google Books.
Esse foi o resultado de seis meses de pesquisa e dedicação. Junto com o primeiro livro sobre Acessibilidade na Web, foi um ano de trabalho para escrever duas publicações inteiras para um curso universitário. Uma grande responsabilidade, mas que me deixa feliz pelo resultado alcançado.
Esse livro aborda a Web em diversos aspectos, para que possamos conhecer um pouco da sua história e vislumbrar como ela será dos próximos anos. Nesta publicação são apresentadas as principais tendências da web, além de um pouco da sua história, desde a evolução da linguagem de marcação até a chegada da web semântica e da web de dados. Também são abordadas as novas formas de consumo de conteúdo na web, como conteúdo multimídia, publicações digitais e até dispositivos conectados à internet das coisas, tratando ainda de temas como pagamentos online, segurança e privacidade.
Esse livro tem um papel de trazer mais perguntas do que respostas. A Web de amanhã está em nossas mãos.
Espero que gostem!
Category: livro web Tags: livro, tendências da Web
Pesquisar
Tweets @reinaldoferraz
RT @cewebbr: Fim do 1º lote! Último dia para garantir ingressos para a #Webbr2019 no valor promocional. A Conferência reunirá es... https://t.co/fZxMEnSLlm
5 meses ago
@sandromaia hahahaha muita! Ele foi pensado para isso!
5 meses ago
Esse game mostra o quanto UX é importante para o seu projeto. Dá para entender porque o usuário fica com raiva e de... https://t.co/41SDEAYiBY
5 meses ago
O Workshop sobre Web Games do W3C está sendo transitido online.
https://t.co/ueyNnbDyDR
6 meses ago
Seguir
Reinaldo Ferraz
Reinaldo Ferraz
Arquivos
abril 2018
dezembro 2017
fevereiro 2016

Figura 7.2: Imagem de um site não responsivo

Se isso não for o suficiente, lembre-se de que desde 2015 o Google adotou política de exibição de páginas que pune sites não responsivas (https://exame.abril.com.br/tecnologia/google-deixa-de-mostrar-nas-buscas-sites-nao-responsivos/).

Qual a barreira de acesso? Sites não responsivos já são ruins para ler, navegar e interagir para pessoas sem deficiência que utilizam dispositivos móveis. Para pessoas com deficiência essa dificuldade acaba sendo maior, já que a tela pequena vai exibir muita informação de forma desproporcional, amontoada e com textos e links muito pequenos.

7.2 ORIENTAÇÃO

Ainda relacionado ao design responsivo, é importante que o usuário não seja impedido de alterar a orientação da tela do dispositivo móvel, desde que a aplicação não dependa exclusivamente de determinada orientação (por exemplo, jogos online que necessitam da orientação horizontal).

Em CSS, considere sempre a reorganização de conteúdo que leve em conta os modos horizontal e vertical. É possível customizar cada um deles utilizando seletores específicos, como os seguintes:

```
@media only screen and (orientation:portrait){}
@media only screen and (orientation:landscape){}
```

O importante é garantir que o usuário tenha flexibilidade para usar a aplicação da forma que achar melhor. Por isso, o design responsivo tem papel importante na acessibilidade.

QUAL A BARREIRA DE ACESSO? Pessoas com baixa visão e mobilidade reduzida podem escolher orientação horizontal para ler com mais facilidade ou interagir com o conteúdo que pode ser desconfortável na orientação vertical.

7.3 USO DE CORES

A relação das cores com a Web vem desde os seus primórdios. Aprendemos a utilizar cores com atributos como `color` e `bgcolor`, que hoje não devem ser mais usados na página e sim no CSS. O fato é que as cores têm um papel importante nas páginas Web, mas devem ser utilizadas com cuidado.

Tanto pessoas com baixa visão quanto daltônicos podem ter dificuldade em enxergar uma determinada cor ou um contraste inadequado. Fontes pequenas com cores claras em fundo branco cria uma séria barreira de acesso.

Por isso, é importante termos alguns cuidados com o uso de cores nas páginas:

Nunca use somente a cor para transmitir uma informação

Esse é um mantra que deve ser repetido desde a concepção do projeto. A questão aqui não é proibir o uso das cores, mas sim permitir que a informação disponível em cor esteja acessível para as pessoas que não enxergam ou não percebem todas as cores.

Por exemplo, imagine que você desenvolveu uma aplicação com uma tabela de horários para os ônibus. Nessa tabela são exibidos os ônibus que estão no horário e os que estão atrasados. Você criou ícones de um ônibus vermelho para representar os atrasados e verde para representar os que estão no horário. Porém, quando exibidos para uma pessoa que não enxerga uma dessas duas cores, elas podem parecer algo assim:

Figura 7.3: Ícones de dois ônibus em cinza

Para evitar esse tipo de problema você pode utilizar marcações nas imagens para que o usuário que não enxerga as cores saiba o que significa cada ônibus, como no exemplo a seguir:

Figura 7.4: Ícones de dois ônibus em cinza. O primeiro tem a marca de um X e o segundo uma marca de OK

E em vez de usar a referência por cor, utilize os símbolos para identificar os ônibus. Em vez de “ônibus em vermelho estão atrasados", utilize “ônibus marcados com X estão atrasados" e use o vermelho para reforçar isso de forma visual para quem enxerga.

Importante: não esqueça de descrever a imagem no atributo `ALT` ou em texto (caso a imagem venha de um CSS):

```
<img src="images/onibus11.png" alt="ônibus atrasado">
```

O uso de ícones/imagens para complementar a informação apresentada em cores também ajuda pessoas com deficiências cognitivas ou de aprendizagem, como pessoas disléxicas ou autistas, pois fornecem pistas visuais para encontrar informações mais facilmente, ajudam escanear melhor o conteúdo e facilitam a memorização por associar palavras ou termos com um conteúdo visual. Referência: https://blog.fullfabric.com/how-to-design-visual-learning-resources-curriculum-for-neurodiverse-audience-autism-autistic

> **Qual a barreira de acesso?** Informações disponíveis somente em cor dificultam o acesso de pessoas que não enxergam um determinado espectro de cor, seja de uma cor específica ou de todas as cores. Utilizando essas técnicas você garante que o usuário que não enxerga cores vai conseguir compreender a informação transmitida.

Cores de texto e cores de fundo

Existe uma relação de contraste mínima entre o fundo da página e o texto para que ele possa ser lido principalmente por pessoas com baixa visão. Esse contraste pode variar dependendo do tamanho do texto:

- Textos grandes requerem um contraste de 3:1

- Textos normais requerem contraste de 4,5:1
- Textos pequenos requerem contraste de 7:1

Mas não é necessário se matar para fazer cálculos para saber se a relação de contraste está adequada. Existem ferramentas que facilitam esse trabalho, como o Color Contrast Checker (https://webaim.org/resources/contrastchecker/) da WebAim e plug-ins para navegadores, como o Color Contrast Analyser (https://chrome.google.com/webstore/detail/color-contrast-analyzer/dagdlcijhfbmgkjokkjicnnfimlebcll) para o Google Chrome e o WCAG Contrast Checker (https://addons.mozilla.org/pt-BR/firefox/addon/wcag-contrast-checker/) para Mozilla Firefox. Existem outras ferramentas e plug-ins disponíveis na internet. Utilize o que achar mais confortável e adequado ao seu projeto.

Essa regra de contraste funciona mas também deve ser verificado se a combinação de cores não tem uma barreira para pessoas que não enxergam determinados expectros de cores, como pessoas com daltonismo. Em certos casos, os verificadores de contraste podem dizer que um texto azul com fundo verde pode passar pelos critérios do WCAG, mas talvez essa combinação cause problemas para quem não enxerga alguma dessas cores. Isso também pode ser uma barreira para pessoas que têm sensibilidade a cores muito brilhantes/saturadas.

Para essa situação também existem ferramentas que auxiliam no processo de verificação de barreiras para pessoas com daltonismo. Uma delas é o ASES - Avaliador e Simulador de Acessibilidade em Sítios disponível para Windows e Linux (https://softwarepublico.gov.br/social/ases). Ele possui ferramentas

para simular diversas características e deficiências relacionadas a baixa visão, desde daltonismo até catarata. Existem também ferramentas online e plugins, um deles é o Color Blind Webpage Filter (https://www.toptal.com/designers/colorfilter/) que mostra como seu site se comporta com diversas variações de daltonismo. O mesmo vale para plugins para navegadores. Uma busca nas extensões do Google Chrome ou Mozila Firefox por "color blind checker" trará diversas opções para implementar no navegador.

Lembre-se de que a verificação de contraste deve considerar os mais diversos cenários. Usuários de computadores em ambientes fechados costumam ter pouca incidência de luz na tela, já usuários de dispositivos móveis podem ter problemas ao ler textos em ambientes abertos, com incidência de sol sobre a tela do seu smartphone.

> **Qual a barreira de acesso?** Contraste em texto serve para garantir que pessoas com baixa visão, seja devido a daltonismo ou por limitações relacionadas com a idade avançada, não tenham barreiras em consumir conteúdo na rede. Textos pequenos, com baixo contraste ou com cores incompatíveis podem dificultar a leitura de um grande grupo de pessoas.

Outras aplicações de cores e contraste

Contraste não deve ser verificado apenas quando existem textos na página. Imagens também podem necessitar de contraste quando elas têm alguma relação com o conteúdo da página.

Essa "relação" com a página pode se referir a diversas situações: um infográfico, um banner ou mesmo uma figura que ilustra uma situação que complementa a informação em texto da página. Imagens que têm apenas função decorativa, como bullets ou molduras, não entram nessa categoria de imagens que transmitem informações ao usuário.

Se a imagem transmite informação ou interfere na compreensão do conteúdo da página, suas cores devem seguir a mesma relação de contraste entre o texto e o plano de fundo. Textos em imagem são desaconselhados, mas se for essencial para a aplicação ele deve seguir as mesmas orientações de contraste de texto.

Logotipos não têm requisitos de contraste. Já textos que complementam a imagem podem exigir esse requisito.

QUAL A BARREIRA DE ACESSO? Os usuários podem alterar o contraste de uma página com aplicações, plugins ou mesmo em links na própria página, mas esse recurso não consegue interferir no contraste de imagens, principalmente as que não tem fundo transparente. Sendo assim, o contraste é fundamental caso a imagem transmita informação ou faça parte do contexto e necessite ser compreendida pelo usuário.

7.4 REDIMENSIONAR TEXTO

No início deste capítulo eu disse que o design responsivo possui relação com acessibilidade. Pois bem. Este tópico toca em

uma das questões mais importantes relacionadas ao design responsivo.

Segundo as diretrizes do W3C, o usuário tem que ser capaz de redimensionar o texto em até 200% sem perda de funcionalidade ou conteúdo. Isso significa que a aplicação deve se adaptar, sem perda para o usuário, quando este der zoom na página.

Existem algumas formas de dar zoom na página: no computador, por atalhos de teclado (utilizando "ctrl +" ou "command +"), que podem ser customizados para dar zoom em toda a página ou somente no texto. No smartphone o usuário pode mudar essas configurações no próprio dispositivo, já que o movimento de "pinça" na maioria das vezes faz o zoom de toda a página.

O usuário também consegue redimensionar o texto quando a aplicação oferece esse recurso na própria página. Apesar de ser um recurso controverso (já que os dispositivos já proporcionam uma forma de zoom nativa) é mais um recurso que o usuário pode usar, se disponível.

Bloquear o `viewport` com `content="user-scalable=no"` (utilizado no elemento `<meta>` que fica no início do código HTML, dentro do elemento `<head>`) é ruim para o usuário, já que esse atributo impossibilita dar zoom na página. Evite esse recurso (a menos que esse bloqueio seja fundamental para o funcionamento da aplicação).

Outra boa prática é, na definição do tamanho de fontes, utilizar unidades relativas em vez de absolutas.

Em CSS, em vez de utilizar:

```
font-size: 10px;
```

Utilize:

```
font-size: 1em; ou font-size:100%;
```

O valor em `em` considera o valor anterior definido para a aplicação e aumenta proporcionalmente. Utilizar `1em` significa que você mantém o tamanho do texto definido inicialmente, já utilizar `2em` dobra o tamanho do texto em comparação com o tamanho predefinido.

Além de todas essas recomendações no redimensionamento de texto, tenha certeza de que quando o usuário utilizar zoom de 200% na sua página ou aplicação ele não vai perder conteúdo. O design da página também não deve ser quebrado e os blocos de texto não devem sobrepor elementos interativos ou de conteúdo.

QUAL A BARREIRA DE ACESSO? Usuários com baixa visão costumam navegar com a configuração de fontes maiores que o padrão do dispositivo, ou utilizam esse recurso de zoom para ler páginas com textos pequenos. Se a página não se comportar adequadamente pessoas com baixa visão podem não conseguir utilizar a página ou aplicação.

7.5 REORGANIZAÇÃO DE CONTEÚDO

O documento WCAG 2.1 trouxe essa nova recomendação que coloca um valor limite à largura e à altura mínima sem a necessidade de rolagem em duas dimensões. Nesse caso os valores

seriam para as seguintes situações:

- Conteúdo de rolagem vertical com largura equivalente a 320 pixels (definidas em CSS);
- Conteúdo de rolagem horizontal a uma altura equivalente a 256 pixels (definidas em CSS).

O valor de 320 pixels em CSS equivale a uma largura inicial de visualização de 1280 pixels em 400% de zoom. Para conteúdo da Web projetado para rolar horizontalmente (por exemplo, com texto vertical), os 256 pixels do CSS equivalem a uma altura inicial de `viewport` de 1024 pixels a 400% de zoom.

Isso não se aplica para conteúdo que necessite de rolagem bilateral para sua compreensão.

Alguns exemplos de conteúdo que necessitam de rolagem bidimensional são imagens, mapas, jogos ou tabelas de dados onde o usuário precisa manter uma barra de ferramentas à vista para manipular o conteúdo.

Qual a barreira de acesso? O objetivo desta técnica é evitar barreiras de acesso para pessoas com baixa visão que precisam ampliar o texto e lê-lo em uma única coluna. Quando o zoom do navegador é usado para dimensionar o conteúdo para 400%, ele é reorganizado - ou seja, é apresentado em uma coluna para que não exista rolagem para as duas direções (vertical e horizontal).

7.6 ANIMAÇÕES DE INTERATIVIDADE

No caso de animações que são disparadas a partir da interatividade do usuário, deve existir um dispositivo para parar esse tipo de animação. Isso possibilita que o usuário controle o conteúdo animado que pode interferir na página baseado em sua interação.

Essa recomendação se aplica apenas se a não interrupção das animações seja essencial para a funcionalidade ou compreensão da informação.

QUAL A BARREIRA DE ACESSO? Pessoas com deficiências cognitivas podem ter dificuldade para operar e compreender o conteúdo (considerando que podem ter velocidades de leitura diferentes) de uma aplicação com uma animação em movimento que não pode ser parada. Animações automáticas geram distrações que atrapalham pessoas que possuem dificuldade de foco e concentração. O usuário precisa desse recurso para poder consumir o conteúdo ou utilizar a aplicação.

7.7 CONTEÚDO QUE CAUSE CONVULSÕES

A interface de uma aplicação Web é na maioria das vezes visual. Isso significa que certas pessoas podem estar expostas a conteúdos que piscam em determinadas frequências que podem causar distúrbios (como convulsões).

Sendo assim, a recomendação é que evite conteúdos que piscam muito rápido em grandes áreas da tela. A documentação do W3C é bem detalhada nesse aspecto (mostrando a relação de área com frequência de movimento, por exemplo), mas desenvolver conteúdo que não pisque rapidamente em grandes áreas pode evitar que usuários do site sejam surpreendidos por uma animação que cause desconforto.

Qual a barreira de acesso? Pessoas com fotossensibilidade, sensibilidade sensorial ou que possuem condição de epilepsia podem sentir desconforto ou até passar mal em sites com grandes áreas piscando na tela.

7.8 ESCONDENDO E EXIBINDO CONTEÚDO

Quando trabalhamos com acessibilidade (principalmente descrição de um determinado conteúdo) é comum lidarmos com situações onde queremos esconder conteúdo exibido no navegador e permitir que seja acessível por leitores de tela, ou exibir conteúdo que deve ficar invisível para tecnologia assistiva.

Podemos fazer isso usando recursos simples de esconder e exibir conteúdo utilizando CSS. E para isso temos alguns atributos bem conhecidos.

Na tabela a seguir apresentamos como cada um deles trata conteúdo escondido e sua relação com leitores de tela:

CSS	Efeito na tela	Acessibilidade
`visibility:hidden;`	O objeto fica oculto mas ocupa espaço na tela.	O conteúdo não é lido por leitores de tela.
`display:none;`	O elemento fica oculto e não ocupa espaço na tela.	O conteúdo não é lido por leitores de tela.
`height: 0; width: 0; overflow: hidden;`	O elemento fica oculto e não ocupa espaço na tela.	Implementação não padronizada. Alguns leitores leem, outros não.
`text-indent: -99999em ;`	O elemento é apenas movido para fora do `viewport` mas elementos interativos permanecem acessíveis por teclado.	O conteúdo em texto é lido por leitores de tela.
`position: absolute; left: -999em;`	O elemento é apenas movido para fora do `viewport` mas elementos interativos permanecem acessíveis por teclado.	O conteúdo é lido por leitores de tela.

Esta tabela foi baseada no documento eMag e está disponível no site da documentação, em http://emag.governoeletronico.gov.br/#r1.5.

Qual a barreira de acesso? Algumas informações que ficam escondidas dos usuários podem (ou não) ser lidas por usuários de leitores de tela. Utilizar a técnica adequada permite definir o que o usuário vai ler ou não utilizando tecnologia assistiva.

Capítulo 8

ELIMINANDO BARREIRAS - MULTIMÍDIA E CONTEÚDO NÃO TEXTUAL

A Web evoluiu de um conjunto de hiperlinks e texto para uma aplicação robusta, que suporta interatividade avançada e conteúdo audiovisual. Quando abordamos o tópico relacionado a conteúdo multimídia precisamos considerar o acesso de pessoas que podem não conseguir ouvir ou ver o conteúdo apresentado. Neste capítulo serão abordadas as principais formas de tornar acessíveis os elementos audiovisuais, como imagens, sons e vídeo.

8.1 DESCREVENDO IMAGENS

Apesar de ser algo tecnicamente simples, descrever imagens na Web demanda atenção. Muitas vezes a descrição da imagem pode não representar o que a figura quer dizer ou está em um lugar diferente na página, que o usuário precisa identificar para relacionar com a imagem.

Para publicar imagens com descrições na Web você pode

utilizar as seguintes técnicas:

Utilizando o atributo ALT

Este talvez seja o mais poderoso e importante atributo para descrever imagens na Web. Basta utilizar o atributo `alt` dentro de um elemento `<img>` e proporcionar a descrição:

```
<img src="img/foto.jpg" alt="Ursinho de pelúcia">
```

As boas práticas para descrever imagens na Web envolvem principalmente a verificação humana, para determinar se aquele texto realmente representa aquela imagem e como descrever o conteúdo de uma forma simples. A primeira ação a tomar é se perguntar: qual a função desta imagem na página? Para isso vamos tentar identificar como ela se enquadra nas categorias a seguir:

Ela é informativa?

Se a imagem faz parte do contexto da página ou transmite informação ao usuário (seja ela uma foto ou ilustração) ela precisa de uma descrição. Por exemplo:

```
<h2>Bombeiros salvam animais em perigo</h2>
<img src="img/08653.jpg" alt="Bombeiro resgatando um filhote de g
ato no incêndio na Califórnia">
<p>O corpo de bombeiros resgatou diversos animais que estavam pre
sos em uma fazenda durante o incêndio que destruiu a região ao su
l da Califórnia...</p>
```

Se a imagem é parte de uma informação, como um ícone, não precisa ser descrito. O mais importante é a informação que ele precisa passar. Imagine um ícone de contato, que mostre uma imagem de um telefone. Não faz sentido descrever no atributo `alt` que aquilo é uma "ilustração de um telefone fora do gancho".

Nesse caso basta descrevê-lo da seguinte forma:

```
<img src="img/telefone-icone.jpg" alt="Telefone:"> 999-9999
```

Se o número do telefone está no ícone, é necessário colocá-lo dentro do atributo `alt`:

```
<img src="img/telefone-icone.jpg" alt="Telefone: 999-9999">
```

Mas se a imagem é um ícone de um formato de arquivo, por exemplo, para fazer o download de uma publicação em PDF ou em Word, a descrição precisa ser feita no ícone:

```
<a href="arquivo.doc"><img src="img/word-icone.jpg" alt="Document
o Word">(100 KB)</a>
<a href="arquivo.pdf"><img src="img/pdf-icone.jpg" alt="Documento
 PDF">(1.2 MB)</a>
```

O mesmo vale para um ícone para abrir uma nova janela:

```
<a href="http://www.reinaldoferraz.com.br" target="_blank">Blog d
e Reinaldo Ferraz <img src="img/word-icone.jpg" alt="Abre em uma
nova janela"></a>
```

Não é necessário escrever "Foto de …" ou "Ilustração de…" quando descrever uma imagem no atributo `alt`. Essa informação já é passada para o usuário pelo leitor de telas quando ele acessa o elemento, que o identifica como "imagem". Também é importante ser sucinto na descrição da imagem. Lembre-se de que quem está consumindo essa imagem está ouvindo um áudio de uma parte do conteúdo de uma página Web. O mais importante é que ela seja descrita baseada em seu contexto, mas de forma simples. Sendo assim, em vez de descrever:

```
<img src="img/foto234.jpg" alt="Foto de um ursinho de pelúcia mar
rom usando uma gravata borboleta vermelha com bolinhas brancas. E
le está sentado na grama verde sob a luz do sol">
```

Seja breve:

```
<img src="img/foto234.jpg" alt="Ursinho de pelúcia">
```

Claro que tudo depende do contexto. Se essa foto foi feita para vender o ursinho de pelúcia os detalhes são válidos. Tudo tem que estar ligado ao seu contexto.

Ela é apenas decorativa?

Uma boa prática no desenvolvimento de aplicações Web é colocar imagens decorativas somente dentro do CSS e não dentro do HTML da página. Caso não seja possível levá-la para dentro do CSS, o ideal é deixar o atributo `alt` em branco. Assim os leitores de tela podem ignorar essa imagem.

```
<img src="img/separador.jpg" alt="">
```

Essa técnica não costuma ser recomendada porque nem todos os leitores de tela ignoram a figura. Alguns deles identificam que existe uma figura na página. Para evitar esse tipo de situação utilize o atributo `role="presentation"` na figura.

```
<img src="img/separador.jpg" alt="" role="presentation">
```

O atributo `role="presentation"` tira toda a semântica do elemento, fazendo com que ele se comporte como um bloco genérico. Se esse bloco não tem conteúdo dentro do atributo `alt` os leitores de tela não leem esse elemento.

Ela é funcional?

Imagens funcionais são aquelas que servem principalmente como recurso de interação, sejam elas para links ou para botões.

É muito comum e uma boa prática ter imagens como o logo do

website para retornar a página inicial. Nesse caso não é necessário descrever o logotipo. Basta colocar a informação para "voltar para a página inicial" no atributo `alt`:

```
<a href="/"><img src="img/logo.png" alt="voltar para a página pri
ncipal"></a>
```

Descrever o logo não é obrigatório, mas pode ser uma boa prática se ele for descrito em uma página específica (por exemplo, em uma página sobre a empresa). Descrever o logo em todas as páginas pode deixar o usuário de leitor de tela desconfortável com a descrição repetitiva.

Botões para envio de formulários que utilizam o `<input type="submit">` tem a descrição da sua função no atributo `value`. No caso de botões em imagens, utilizando `<input type="image">` é necessário que sua função seja descrita no atributo `alt`:

```
<input type="image" src="img/botao-enviar" alt="enviar">
```

Ela é uma imagem de texto?

Apesar de não ser uma boa prática utilizar imagens com texto, elas ainda existem. Atualmente, é possível manipular as imagens com CSS com um resultado muito melhor do que uma imagem. Basicamente o conteúdo do atributo `alt` deve descrever exatamente o texto, sem detalhes descritivos do formato ou posição do texto:

```
<img src="img/banner-loja.jpg" alt="A maior loja de eletrônicos d
o Brasil">
```

Ela é uma imagem complexa?

Ser sucinto na descrição de imagens nem sempre é válido

quando se tem imagens complexas que precisam ser descritas. O uso do atributo `alt` não deve servir para fazer a descrição longa. O texto desse atributo deve informar do que se trata a imagem e onde está sua descrição detalhada, que pode ser feita de diversas formas.

Imagine a situação de descrever um gráfico. Esse gráfico tem diversas informações em números que devem ser informadas ao usuário. Nesse caso a descrição pode ser feita de algumas maneiras:

Descrição detalhada próxima da imagem

Existem diversas técnicas para relacionar o texto a imagem. Você pode deixar o texto visível para todos os usuários ou mantê-lo escondido e voltar a exibi-lo somente para usuários de leitores de tela utilizando recursos de CSS (como vimos no último tópico do capítulo anterior sobre eliminar barreiras de design).

Uma das técnicas é relacionar a figura ao texto utilizando os elementos `<figure>` e `<figcaption>`:

```
<figure><img src="img/grafico.jpg" alt="Gráfico de vendas. A desc
rição detalhada do gráfico está a seguir">
<figcaption>O gráfico representa o percentual de vendas dos últim
os três meses das filiais de Porto Alegre e João Pessoa. Nos mese
s de março, abril e maio a filial de Porto Alegre vendeu 25%, 35%
 e 44% respectivamente. A filial de João Pessoa vendeu 29%, 27% e
 39% respectivamente"</figcaption>
</figure>
```

Dessa forma, o conteúdo do elemento `<figcaption>` pode ser manipulado com CSS para definir como ele vai ser exibido.

Também é possível utilizar o atributo `aria-describedby` para relacionar a figura com sua descrição complementar, da mesma forma que fizemos anteriormente com campos de

formulário:

```
<img src="img/grafico.jpg" alt="Gráfico de vendas. A descrição de
talhada do gráfico está a seguir" aria-describedby="detalhe">
<p id="detalhe">O gráfico representa o percentual de vendas dos ú
ltimos três meses das filiais de Porto Alegre e João Pessoa. Nos
meses de março, abril e maio a filial de Porto Alegre vendeu 25%,
 35% e 44% respectivamente. A filial de João Pessoa vendeu 29%, 2
7% e 39% respectivamente"</p>
```

Se essa informação complementar estiver escondida fora da tela, lembre-se de utilizar técnicas que permitem que o leitor de tela a acesse, como vimos no capítulo anterior.

Link para a descrição detalhada da imagem

Para não colocar a descrição da imagem na mesma página, é possível proporcionar um link que direciona o usuário para a descrição detalhada da imagem. Assim não é necessário considerar o texto da descrição dentro da página:

```
<img src="img/grafico.jpg" alt="Gráfico de vendas. A descrição de
talhada do gráfico está no link a seguir">
<a href="detalhe.htm">Link para a descrição detalhada do gráfico<
a>
```

É importante lembrar que esses exemplos de descrição de imagens permitem que ferramentas de busca indexem o conteúdo das descrições (exceto o conteúdo dentro do atributo `aria-describedby`).

Ela faz parte de um grupo de imagens?

É comum que aplicações usem imagens sequenciais ou agrupadas em determinadas situações. Por exemplo, uma forma de exibir um ranking de 5 estrelas ou uma galeria de fotos. Nesses casos a recomendação é pensar na leitura do conjunto como um

todo, isto é, como aquelas imagens serão lidas em sequência, e não individualmente.

No caso do ranking, você pode considerar a primeira imagem como a que vai transmitir a informação ao usuário. Por exemplo, em um conjunto de cinco estrelas mostrando a média de satisfação com um produto:

```
<p>Nível de satisfação:</p>
<img src="img/star-cheia.png" alt="4 de 5 estrelas">
<img src="img/star-cheia.png" alt="">
<img src="img/star-cheia.png" alt="">
<img src="img/star-cheia.png" alt="">
<img src="img/star-vazia.png" alt="">
```

Ela faz parte de um mapa de imagem?

Mapas de imagens são aquelas áreas interativas dentro de uma única imagem. Neste caso a descrição da imagem deve servir de tal maneira que o usuário consiga tanto identificar a imagem por completo quanto cada área do mapa, descrevendo cada uma delas.

Isso é muito comum em imagens que mostram organogramas de empresas ou mapas geográficos interativos. Neste caso, o atributo `alt` da imagem deve representar a imagem por completo, como no exemplo a seguir:

```
<img src="img/mapa-brasil.jpg" alt="Mapa do Brasil" usemap="#mapa
>
<map id="mapa" name="mapa">
<area shape = "poly" alt="São Paulo" coords = "414, 706, 188, 19,
6, 267, 674" href="/saopaulo/>
```

Tudo o que foi apresentado aqui representa as boas práticas para tornar o conteúdo mais amigável para usuários que não conseguem enxergar imagens. É notável que essa intervenção é tecnicamente simples, mas depende de um bom estudo de como

essa figura vai ser descrita dentro de uma página ou aplicação. Ela depende muito mais de interpretação de contexto do que habilidade técnica com HTML.

QUAL A BARREIRA DE ACESSO? Imagens são referências visuais e precisam de alternativas textuais para que usuários de leitores de tela consigam compreender o seu significado. Não descrever uma imagem significa que estamos ignorando o acesso das pessoas que não conseguem enxergar uma imagem.

NOTA IMPORTANTE: Texto alternativo dentro do atributo `alt` é indexado pelo Google e pode ser encontrado em uma pesquisa por texto. Esse mesmo conteúdo ajuda o buscador a relacionar a imagem com sua descrição na busca por imagens (http://ceweb.br/media/docs/publicacoes/19/acessibilidade-explorando-atributos-web.pdf).

8.2 DESCREVENDO IMAGENS EM SVG

Diferente da forma como publicamos imagens em bitmap na Web, o uso de vetores já é permitido diretamente no HTML sem o uso de um elemento `img` .

Isso não impede o uso de `<img src="img/star.svg" alt="estrela">` para exibir uma imagem vetorial em uma

aplicação, da mesma forma que inserimos uma imagem em bitmap. A vantagem de aplicar os objetos vetoriais dentro do HTML é a possibilidade de manipulação de cada elemento do código. O exemplo a seguir mostra o código de uma imagem SVG embutida dentro do HTML de uma página:

```
<svg xmlns="http://www.w3.org/2000/svg" width="200" height="200"
id="svg01">
<g transform="translate(0,-852.36218)">
<path style="fill:#ff0000;stroke:#9b0000;stroke-width:10;" d="m 1
44.10177,103.8611 a 51.799896,51.799896 0 1 1 -103.599794,0 51.79
9896,51.799896 0 1 1 103.599794,0 z" transform="translate(5.69783
74,845.45332)"/>
<path style="fill:#006a92;stroke:navy;stroke-width:10;" d="M 106.
58415,98.958225 69.586242,79.647381 32.703751,99.177762 39.636472
,58.023282 9.6646592,28.981168 50.947225,22.857144 69.306117,-14.
622251 87.887424,22.747375 129.20566,28.625949 99.40697,57.845672
 z" transform="translate(28.564552,907.03667)" />
<text style="fill:#ffffff;">
<tspan x="62.024254" y="948.65192">This is a star</tspan>
</text>
</g>
</svg>
```

O resultado deste código gera a seguinte imagem:

Figura 8.1: Estrela

Da forma como essa imagem foi publicada, ela não está acessível. Não existe nenhuma descrição que informa que a figura representa essa estrela. Mas existem alguns elementos e atributos que podem ser inseridos no SVG para tornar isso possível.

Segundo a documentação do W3C, utilizar os elementos `<title>` e `<desc>` seria o suficiente para promover acessibilidade. O problema é que nem todos os leitores de tela/navegadores conseguem ler os dois documentos. Escrevi um post no meu blog que detalha o uso desses elementos em http://www.reinaldoferraz.com.br/acessibilidade-seo-e-svg/.

Para que os leitores de tela consigam perceber descrições no conteúdo SVG inline, é necessário adicionar alguns atributos de ARIA.

Utilizar os atributos `role="img"` e `aria-label="descrição da imagem"` no elemento `<svg>` resolveria o problema, mas todo o restante do código SVG seria compreendido como uma única imagem. Isso pode ser um problema se você tem texto dentro do SVG utilizando o elemento `<text>` . Dessa forma, o leitor de tela vai ignorar o conteúdo do elemento textual.

No nosso exemplo, o recomendável seria utilizar esse atributo em um objeto que não englobe o texto. No caso, o primeiro elemento `<path>` poderia receber essa descrição:

```
<path style="fill:#ff0000;stroke:#9b0000;stroke-width:10;" d="m 1
44.10177,103.8611 a 51.799896,51.799896 0 1 1 -103.599794,0 51.79
9896,51.799896 0 1 1 103.599794,0 z" transform="translate(5.69783
74,845.45332)" role="img" aria-label="Estrela azul sobre um círcu
lo vermelho" />
```

Dessa forma o elemento `<text>` preserva suas características

de contemplar conteúdo textual.

Essa técnica é interessante pois possibilita a descrição de todos os elementos do SVG, como no exemplo a seguir:

```
<path style="fill:#ff0000;stroke:#9b0000;stroke-width:10;" d="m 1
44.10177,103.8611 a 51.799896,51.799896 0 1 1 -103.599794,0 51.79
9896,51.799896 0 1 1 103.599794,0 z" transform="translate(5.69783
74,845.45332)" role="img" aria-label="Círculo vermelho" />

<path style="fill:#006a92;stroke:navy;stroke-width:10;" d="M 106.
58415,98.958225 69.586242,79.647381 32.703751,99.177762 39.636472
,58.023282 9.6646592,28.981168 50.947225,22.857144 69.306117,-14.
622251 87.887424,22.747375 129.20566,28.625949 99.40697,57.845672
 z" transform="translate(28.564552,907.03667)" role="img" aria-la
bel="Estrela" />
```

Apesar de terem sua implementação não padronizada pelos leitores de tela e navegadores, os elementos `<title>` e `<desc>` são indexados por ferramentas de busca (tenho um artigo detalhado sobre como ferramentas de busca indexam o conteúdo desses elementos em http://ceweb.br/media/docs/publicacoes/19/acessibilidade-svg-seo-Cewebbr-ReinaldoFerraz.pdf). Sendo assim, é recomendável que eles sejam utilizados. Ambos são utilizados para descrever a imagem mas não são exibidos nos navegadores. Em alguns navegadores, o elemento `<title>` funciona como um tooltip ao passar o mouse.

Resumindo, um código adequado para essa implementação seria o seguinte:

```
<svg xmlns="http://www.w3.org/2000/svg" width="200" height="200">
<title id="title-star">Escudo de estrela</title>
<desc id="desc-star">Estrela azul sobre um círculo vermelho</desc

<g transform="translate(0,-852.36218)">
<path style="fill:#ff0000;stroke:#9b0000;stroke-width:10;" d="m 1
```

```
44.10177,103.8611 a 51.799896,51.799896 0 1 1 -103.599794,0 51.79
9896,51.799896 0 1 1 103.599794,0 z" transform="translate(5.69783
74,845.45332)" role="img" aria-label="Estrela azul sobre um círcu
lo vermelho" />
<path style="fill:#006a92;stroke:navy;stroke-width:10;" d="M 106.
58415,98.958225 69.586242,79.647381 32.703751,99.177762 39.636472
,58.023282 9.6646592,28.981168 50.947225,22.857144 69.306117,-14.
622251 87.887424,22.747375 129.20566,28.625949 99.40697,57.845672
 z" transform="translate(28.564552,907.03667)" />`
<text style="fill:#ffffff;">
<tspan x="62.024254" y="948.65192">This is a star</tspan>
</text>
</g>
</svg>
```

> **Qual a barreira de acesso?** Da mesma forma que imagens vindas do elemento `<img>` precisam ser acessíveis, conteúdo de SVG inline também precisa ser descrito para usuários de leitores de tela.

8.3 ACESSIBILIDADE EM ÁUDIO

Para garantir a acessibilidade em áudio, o ideal é proporcionar a transcrição em formato de texto. Esse texto pode ficar na própria página ou em um link específico, sem afetar o design da página.

Pode parecer difícil transcrever um texto de um podcast ou de uma entrevista, mas existem técnicas e recursos que podem ajudar nessa tarefa. No caso da entrevista ou de uma notícia muitas vezes temos disponível o roteiro desse material, o que ajuda a não ter que digitar tudo o que foi falado. Em outros casos é possível utilizar ferramentas de reconhecimento de áudio que podem transcrever o conteúdo. Uma delas é o WebCaptioner

(https://webcaptioner.com/) que permite a conversão de áudio para texto. Não é perfeito (ainda), mas ajuda bastante. Ele também funciona para tradução em tempo real, similar a um *closed caption*.

Esse recurso de transcrição possibilita que o usuário que compreende português possa ler o conteúdo quando não puder ouvir o arquivo de áudio. Mas e para pessoas que se comunicam somente por LIBRAS?

A transcrição também é útil, já que o texto em português pode ser selecionado e traduzido pela tecnologia assistiva preferida pelo usuário (como aplicações com avatares). Os aplicativos de avatares de LIBRAS estão se popularizando e ficando cada vez mais inteligentes, conseguindo traduzir textos cada vez mais complexos e melhorando a comunicação da comunidade surda.

Um recurso também possível é a inserção de vídeo ou aplicações de avatares que traduzem automaticamente para LIBRAS os textos das próprias páginas.

Segundo a recomendação do WCAG, proporcionar transcrição do áudio é prioridade AA (prioridade maior) em comparação com proporcionar interpretação de língua de sinais no site, que é nível AAA.

QUAL A BARREIRA DE ACESSO? Da mesma forma que devemos nos preocupar em oferecer alternativa textual para quem não consegue enxergar uma imagem, devemos considerar o acesso de pessoas que não conseguem escutar um conteúdo na Web, seja devido à deficiência (como surdez ou baixa audição) ou porque as pessoas não têm disponíveis recursos para ouvir o som em um determinado momento (seja ele em uma biblioteca ou ambiente que requer silêncio).

8.4 ACESSIBILIDADE EM VÍDEO

Além das técnicas sugeridas no item de áudio, o uso de recurso audiovisual em sites tem uma questão adicional, que é a relação do texto com o que é exibido na imagem.

Em vídeos relacionados a treinamentos, por exemplo, proporcionar a transcrição do vídeo pode ajudar pessoas que não escutam. Já para pessoas cegas essa descrição tem que estar relacionada com o conteúdo apresentado, descrevendo as cenas e elementos do vídeo em formato texto.

Essa é a função da audiodescrição, um canal de áudio que utiliza as pausas do vídeo para descrever para o usuário as cenas que vão acontecendo.

Até o momento (junho de 2019) ainda não é possível utilizar o objeto `AudioTrack` para adicionar um canal de audiodescrição, ou mesmo para a descrição em texto (utilizando o atributo `kind="description"` no elemento `<track>`). Existem alguns

experimentos nesse sentido mas a implementação por tecnologia assistiva e browsers ainda não está estável.

Nesse caso o ideal é proporcionar um novo vídeo com audiodescrição para o usuário.

Já na implementação de legendas (que também é um recurso importante de acessibilidade) seu suporte por navegadores vem crescendo a cada dia, principalmente para legendas em formato WebVTT. O formato WebVTT possibilita a sincronização do vídeo com um arquivo em texto (codificado em UTF-8) com as marcações de início e fim de cada bloco. Esse formato é muito simples de ser implementado e é compatível com a maioria dos serviços de publicação de vídeos, como YouTube e Vimeo. Além de possibilitar que o usuário ligue ou desligue as legendas, é possível escolher o idioma da legenda de uma lista disponível.

```
<video controls="true" autoplay="false">
<source src="filme.mp4" type="video/mp4">
<source src="filme.webm" type="video/webm">
<track label="Português" kind="subtitles" srclang="pt" src="legen
da.vtt" default>
<track label="English" kind="subtitles" srclang="en" src="subtitl
e.vtt">
</video>
```

Neste exemplo, estamos entregando para o usuário duas possibilidades de legenda, sendo que a versão em português será entregue por padrão devido ao atributo `default` . Agora, para criar a legenda em um arquivo WebVtt o procedimento é mais simples ainda. Basta criar um arquivo texto (em qualquer editor de texto, como o Notepad) com a seguinte estrutura:

```
WEBVTT

1
```

```
00:00:00.000 --> 00:00:05.000
Olá pessoal, Tudo bem? Eu sou o Reinaldo

2
00:00:05.050 --> 00:00:08.000
e hoje vou apresentar técnicas

3
00:00:08.050 --> 00:00:10.000
para melhorar a acessibilidade das suas páginas Web.

4
00:00:10.010 --> 00:00:14.000
A Primeira dica é utilizar a marcação semântica da HTML5.
```

A primeira exigência é que o arquivo comece com o `WEBVTT`. A partir daí é só colocar a sequência de texto, considerando o primeiro bloco de tempo como o momento que a legenda aparece e a seguinte como o momento que ela deve terminar. No nosso exemplo, o primeiro bloco será exibido por 5 segundos. Já o segundo bloco começará 0.05 segundos depois do primeiro e durará 2.01 segundos. O texto que deve aparecer fica logo abaixo da minutagem de cada bloco.

Depois de produzido esse arquivo, basta salvá-lo com a extensão `.vtt` (por exemplo, `legenda.vtt`) e adicioná-lo na pasta adequada. Não esqueça de mudar sua codificação para UTF-8. A maioria dos editores de HTML permitem essa mudança. No Sublime, por exemplo, é muito simples:

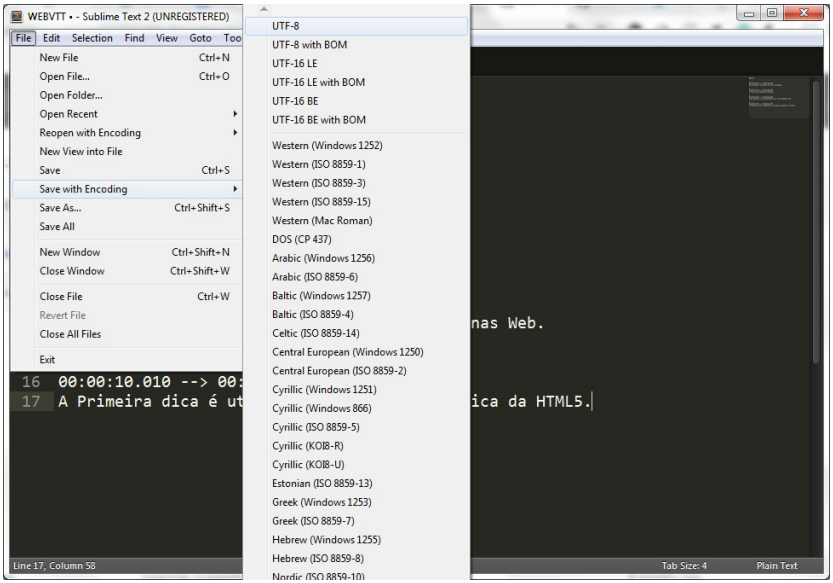


Figura 8.2: Captura de tela do Sublime 3 ilustrando como mudar a codificação de arquivo. Basta ir em File, Save with encode e escolher UTF-8

Também é possível colocar legendas embutidas no vídeo (formato *open caption*) mas essa técnica impede que as legendas sejam desativadas pelo usuário que não necessita desse recurso.

Qual a barreira de acesso? Conteúdos audiovisuais precisam ser acessíveis tanto para pessoas que não escutam quanto para as que não enxergam. Ao proporcionar audiodescrição as pessoas cegas podem ter uma compreensão mais ampla sobre o conteúdo do vídeo, já as legendas servem para garantir que os vídeos sejam acessíveis para as pessoas surdas.

8.5 CONTROLES DE ÁUDIO E VÍDEO

A principal recomendação para controle de elementos multimídia é possibilitar que o usuário pause ou interrompa sua execução. Um áudio que toca automaticamente pode não só atrapalhar a navegação como pode causar constrangimento em um ambiente aberto.

O mesmo serve para elementos em vídeo. Os controles devem permitir que o usuário pause, pare a execução do vídeo e interrompa o áudio.

Apesar da recomendação não englobar vídeos decorativos de fundo, eles podem causar desconforto para algumas pessoas e devem ter uma forma de pausa ou interrupção pelo usuário.

QUAL A BARREIRA DE ACESSO? Áudio e vídeo executado automaticamente pode prejudicar a compreensão do conteúdo ou tirar o foco de leitura/compreensão.

CAPÍTULO 9

ACESSIBILIDADE EM CMS, FRAMEWORKS E OUTRAS APLICAÇÕES PARA DESENVOLVIMENTO WEB

Foi-se o tempo em que desenvolvíamos uma aplicação Web do zero, a partir de documentos `.html`. `.css` e `.js` em branco para criar toda a interface, interação e comportamento de uma aplicação com o seu próprio código. A Web evoluiu e fica muito difícil criar hoje um sistema de e-commerce do zero sem o auxílio de alguns componentes previamente criados para nos ajudar nessa tarefa.

Hoje em dia existem diversas bibliotecas, frameworks e CMSs que nos ajudam a desenvolver uma aplicação sem ter que reinventar a roda. Em um mundo com prazos apertados e aplicações gigantes, essas ferramentas são essenciais para a Web, mas precisamos ficar atentos aos seus recursos de acessibilidade.

Esse tipo de tecnologia vem evoluindo constantemente. Novos frameworks surgem a uma velocidade incrível. Existe até um site que faz uma piada com a quantidade de novos frameworks

(https://dayssincelastjavascriptframework.com/). Mas a questão importante deste livro é: eles estão preparados para garantir que a aplicação final seja acessível?

O fato é que boa parte delas possuem recursos de acessibilidade e seguem as orientações descritas nos capítulos anteriores sobre eliminação de barreiras. Porém, a grande barreira não está na ferramenta e sim na forma como vamos manipular essas aplicações. Utilizá-las sem customização é a mesma coisa que comprar uma TV e colocá-la na sala sem configuração de canais ou aplicativos. Por isso, quando utilizar alguma das ferramentas listadas a seguir, tenha em mente que:

- Os campos de formulário necessitam ser relacionados;
- A estrutura semântica deve ser preservada;
- Não deve haver barreiras de teclado;
- As imagens devem ser descritas adequadamente;
- A aplicação deve ser responsiva;
- O contraste de cores deve ser considerado;
- Todos os elementos interativos devem permanecer acessíveis por teclado.

Essa pequena lista pode ajudar a começar um projeto, mas recomendo repassar as orientações listadas nos capítulos anteriores para efetivamente garantir a acessibilidade da aplicação que está construindo.

9.1 WORDPRESS

Tem uma ampla variedade de temas e plugins, permite customização de praticamente toda a interface de um tema (além

da possibilidade de construção de um tema próprio).

O Wordpress possui uma iniciativa de "Time de Revisores de Tema" e um dos itens que esse grupo verifica é a acessibilidade (https://make.wordpress.org/themes/handbook/review/accessibility/). Temas que passam pelo crivo do grupo de revisores são listados como `accessibility-ready` no diretório de temas (https://wordpress.org/themes/tags/accessibility-ready/).

A documentação de suporte do próprio Wordpress também traz recomendações importantes para publicadores de conteúdo (https://en.support.wordpress.com/accessibility/). Ela explica, por exemplo, como fazer uma boa descrição de imagem e lembra que links devem ser descritos corretamente e não somente com "clique aqui".

Apesar de todo o esforço, nessa mesma página o Wordpress lembra que não pode garantir 100% da acessibilidade do tema. Essa nota é muito importante, pois não basta simplesmente instalar um tema e considerar que ele está acessível.

Existe também uma série de plugins de acessibilidade para Wordpress (https://wordpress.org/plugins/tags/accessibility/) que podem ser instalados em seu site. Alguns deles possibilitam o aumento de fontes, contraste, remoção de atributos redundantes e até players de áudio.

9.2 JOOMLA

Também traz uma série de extensões de acessibilidade (https://extensions.joomla.org/tags/accessibility/) que podem ser utilizadas na aplicação. As extensões vão desde zoom até

sintetizadores de voz que leem o conteúdo da página.

Joomla traz também recursos de acessibilidade na área de administração do CMS. Isso é importante porque uma pessoa com deficiência também é uma publicadora de conteúdo e pode utilizar CMSs para isso. Em sua interface administrativa, o Joomla conta com alguns atributos de `WAI-ARIA` (https://community.joomla.org/blogs/community/wai-aria-roles-in-accessible-admin-template.html) para facilitar a navegação por pessoas com deficiência.

9.3 PLONE

Traz por padrão um número de landmarks de `WAI-ARIA` e possibilita a aplicação de outros caso seja necessário. Apesar de utilizar alguns elementos semânticos do HTML5, os elementos `<div>` criados muitas vezes precisam de descrição. Como é possível adicionar *portlets* em qualquer aplicação em Plone, é fundamental fazer testes para verificar se eles contemplam a acessibilidade ou se existem barreiras de acesso.

Apesar da documentação do Plone declarar que ele está em conformidade com os padrões WCAG e ATAG (https://plone.org/accessibility-info/), lembre-se de que as mudanças feitas na estrutura e design podem interferir nessa conformidade.

9.4 REACT

A estrutura base do React permite que sejam incluídos elementos e atributos para melhorar a acessibilidade de uma

aplicação Web, mas são necessários alguns cuidados.

O uso de atributos de `WAI-ARIA` é um bom exemplo. Dentro de um documento HTML comum os valores desses atributos são escritos normalmente em caixa baixa. Quando adicionar um atributo de `WAI-ARIA` em sua aplicação, lembre-se de utilizá-lo em *camelCase*:

```
aria-label={labelText}
```

A semântica do HTML também é um fator importante para a acessibilidade, como explicamos nos capítulos anteriores. Por isso, preste atenção aos elementos gerados por *sniplets*. Neste exemplo, a criação de uma tabela com um elemento `<div>` aparece entre os elementos estruturais de uma tabela, o que pode provocar não só problemas estruturais na página como dificultar a acessibilidade:

```
class Columns extends React.Component {
  render() {
    return (
      <div>
        <td>Tamanho</td>
        <td>Preço</td>
      </div>
    );
  }
}
```

O uso de fragmentos pode ser útil para corrigir esse código, utilizando `<React.Fragment>` no lugar do elemento `<div>` .

```
class Columns extends React.Component {
  render() {
    return (
      <React.Fragment>
        <td>Tamanho</td>
        <td>Preço</td>
      </React.Fragment>
    );
```

```
  }
}
```

Existem também diversas ferramentas que ajudam a verificar se a sua aplicação em React segue boas práticas de acessibilidade. São elas:

- **eslint-plugin-jsx-a11y** (https://github.com/evcohen/eslint-plugin-jsx-a11y):
  plugin que identifica e aplica várias regras de acessibilidade diretamente no seu JSX.
- **react-a11y** (https://github.com/reactjs/react-a11y): semelhante ao plugin anterior, esta é uma ferramenta de análise para incluir na sua aplicação. Isso permite a exibição de alertas no console quando renderizar a aplicação.
- **react-aria-modal** (https://github.com/davidtheclark/react-aria-modal): gerencia funções de atributos `WAI-ARIA` e outras funções, como navegação por teclado e mapeamento de teclas.

9.5 VUE.JS

Fazer uso de Diretivas Personalizadas pode facilitar e até poupar esforços do usuário em determinadas situações, como adicionar foco automático em um campo de formulário quando a página é carregada, por exemplo. Essa situação permite analisarmos questões importantes de acessibilidade.

Imagine que este formulário seja referente a uma página de solicitação de reembolso de um convênio médico. Para uma pessoa com mobilidade reduzida, o foco automático no primeiro campo

(por exemplo, nome) facilita o acesso. Porém, para um usuário de leitores de tela, o foco automático no primeiro campo pode tornar seu acesso confuso (afinal, ele entrará em uma página diretamente no campo a ser preenchido). Nesse caso, lembre-se de fazer a marcação adequada com os elementos `<fieldset>`, `<legend>`, `<label>` e verifique também a necessidade de colocar uma informação adicional relacionada ao primeiro campo, utilizando o atributo `aria-describedby` relacionado ao bloco de instruções. O resultado do código compilado deve trazer os elementos desta forma:

```
<form>
  <fieldset>
    <legend>Solicitação de Reembolso</legend>
    <label for="name">Nome</label>
    <input type="text" name="name" id="name" aria-describedby="in
structions">
    <div id="instructions">O foco do teclado/cursor está no prime
iro campo do
      formulário de solicitação de reembolso. Inicie o preenchime
nto a partir
      deste campo"</div>
      ... continuação do formulário
```

O elemento `<div>` pode ser escondido utilizando CSS de forma que fique acessível somente para usuários de software leitor de tela. Ele também pode ser retirado se as informações relacionadas ao campo estiverem claras quando o foco estiver no elemento.

Este é um pequeno exemplo, mas que serve para diversas situações que envolvem principalmente a alteração de contexto da página sem a intervenção do usuário.

A semântica também deve ser respeitada na utilização de elementos utilizando Vue.js. Como é possível criar elementos com

as mais diversas funções, é importante ter cuidado para manter a semântica dos elementos.

Por exemplo, por que criar um checkbox utilizando um elemento de lista `<li>` e adicionando diversos atributos de ARIA para torná-lo acessível se temos à disposição elementos de formulário como `<input type="checkbox">`? Entendo que certos projetos e situações podem demandar um novo elemento para algo específico e para isso mesmo existem os atributos de `WAI-ARIA` (neste exemplo, os atributos `role="checkbox"` e `aria-checked="true"`), mas se for possível, prefira o uso de elementos nativos do HTML.

O repositório da iniciativa de acessibilidade Vue-a11y ainda tem poucos recursos, porém alguns são interessantes, como uma forma de implementar salto para conteúdo na aplicação. O repositório está disponível em https://github.com/vue-a11y/.

9.6 ANGULAR

A documentação sobre acessibilidade em Angular (https://angular.io/guide/accessibility/) orienta como adicionar atributos de `WAI-ARIA` na aplicação. O exemplo da documentação é bem simples e intuitivo.

Ao adicionar atributos ARIA em Angular, você deve usar o prefixo `attr` no atributo:

```
<button [attr.aria-label]="myActionLabel">...</button>
```

O resultado é o seguinte:

```
<button aria-label="Salvar documento">...</button>
```

Lembre-se de que valores nos atributos HTML são em caixa baixa. Já propriedades Angular são em *camelCase*.

Recursos de acessibilidade para Angular Material estão disponíveis pelo Angular CDK (Component Dev Kit). É um pacote de componentes com os quais é construída a maioria das ferramentas de Angular. Dentro do repositório estão diversos componentes que podem ser utilizados em um projeto em Angular (https://github.com/angular/components/tree/master/src/cdk/).

Um desses pacotes é o de acessibilidade (https://github.com/angular/components/tree/master/src/cdk/a11y/) que traz uma série de recursos para tornar a aplicação mais acessível para o usuário. A própria documentação do CDK também traz orientações sobre acessibilidade da sua aplicação (https://material.angular.io/cdk/a11y/overview/).

9.7 EMBER.JS

A documentação de acessibilidade do Ember.js (https://guides.emberjs.com/release/reference/accessibility-guide/) apresenta algumas técnicas e plugins que podem ajudar no desenvolvimento de uma aplicação acessível. Apesar de orientar sobre o uso de landmarks de `WAI-ARIA` para as áreas semânticas do HTML5 (o que acaba tornando a informação redundante) a lista de plugins do documento é bastante rica.

Essa documentação também mostra como combinar leitores de tela e navegadores para testes mais efetivos de acessibilidade.

Outra boa documentação que lista alguns recursos de acessibilidade é o blog do Ember.js

(https://blog.emberjs.com/2018/06/17/ember-accessibility-and-a11y-tools.html/). Além de listar ferramentas para facilitar o desenvolvimento acessível, ele explica como adicionar verificações de acessibilidade durante testes de renderização.

9.8 JQUERY

Existem alguns plugins para acessibilidade em JQuery (https://plugins.jquery.com/tag/accessibility/). Esses plugins vão desde uma forma simples de garantir a acessibilidade por teclado até uma forma de verificar se a acessibilidade está sendo contemplada no projeto.

As principais recomendações relacionadas a acessibilidade em JavaScript se aplicam para JQuery, como eventos de mouse que devem também ser considerados via teclado (como `onmouseover` combinado com `onfocus`, por exemplo).

9.9 O QUE MAIS PRECISO SABER?

Todos os recursos listados neste capítulo trazem um ganho enorme para a acessibilidade do projeto, independente de qual deles você está utilizando. O mais importante a se considerar nesse cenário é que não existe "bala de prata" para a acessibilidade. Adicionar um plugin ou mesmo uma técnica que melhora a acessibilidade precisa ser verificada e testada para garantir que ela atinge seus objetivos.

Existem outras ferramentas e bibliotecas que podem ser utilizadas em seu projeto que não foram listadas aqui. A melhor forma de verificar se ela contempla recursos de acessibilidade é

buscar por *accessibility* em sua documentação e entender como ela deve ser implementada. Mesmo que a documentação não aborde a acessibilidade (seja por plugins ou por atributos de `WAI-ARIA`) ela deve permitir a customização para uma eventual intervenção para melhorar a acessibilidade da aplicação.

Capítulo 10

COLOCANDO ACESSIBILIDADE NA ROTINA DO DESENVOLVIMENTO

Depois de tantos capítulos sobre técnicas para tornar sua aplicação acessível, chegou a hora de estabelecermos isso desde o início do projeto. É muito mais fácil quando a acessibilidade é contemplada na rotina do desenvolvimento do que quando a verificamos somente depois da aplicação pronta.

Todo o time deve estar envolvido com a acessibilidade. Deixar a acessibilidade a cargo de um único profissional pode criar gargalos e dificultar a comunicação entre os times. Se todos (designer, programadores, gestores e produtores de conteúdo) estiverem com a acessibilidade em mente durante suas atividades, a chance de o projeto contemplar mais recursos acessíveis é maior.

Costumo comparar a construção de uma aplicação/site com a de uma casa. Se pensar a acessibilidade desde o início do projeto tudo fica muito mais rápido e barato. Colocar uma rampa na entrada de uma casa é mais simples quando contemplada no início

da construção do que no final, quando a casa está pronta. Quebrar degraus custa mais caro, dá mais trabalho e faz mais sujeira. O mesmo acontece com uma aplicação Web. Se tivermos que fazer alterações estruturais em um site já finalizado, o trabalho será maior e isso pode implicar em custos no desenvolvimento.

Por isso decidi adicionar este capítulo para identificarmos as técnicas que devem ser consideradas na rotina do desenvolvimento, desde a concepção do site até o seu desenvolvimento.

10.1 PLANEJAMENTO

Uma aplicação não é construída de uma hora para outra. Existe todo um processo de maturação da ideia até ela ir para a primeira etapa que é o papel. Considerando que o projeto começou do zero, com um novo produto, podemos começar o mais distante possível da aplicação, pensando até mesmo na construção do logotipo.

O documento WCAG não exige contraste entre as cores de um logotipo, mas se você quer que sua marca seja compreendida por um número maior de pessoas vale a pena considerar esta recomendação.

Seguindo para a elaboração de um wireframe, considerando todas as áreas e interações do usuário, leve em conta as diversas situações de uso dessa aplicação com relação à forma de acesso e ao tipo de deficiência do usuário. No checklist a seguir enumeramos algumas boas práticas e apresentamos quem são os beneficiados em cada uma delas.

É fundamental entender como as ferramentas que o usuário

utiliza, como tecnologia assistiva, funcionam para lidar de forma mais efetiva e compreender o impacto que as intervenções na página têm. Para isso, recomendo a leitura do capítulo 2 deste livro, que aborda a forma como tecnologia assistiva funciona.

Coloque-se no lugar do usuário!

Examine seu planejamento e avalie a estrutura criada até o momento. Pense como uma pessoa diferente de você deve acessar aquela aplicação. Será que ela vai navegar com uma ponteira na boca ou com um leitor de tela habilitado? Já pensou em criar *personas* com deficiência para os estudos de UX/UI?

Quanto antes exercitarmos a empatia na construção de uma aplicação, mais fácil será o processo de implementação da acessibilidade.

10.2 DESENVOLVIMENTO

Depois de definida a plataforma de desenvolvimento partimos para a verificação da acessibilidade do que será a base da aplicação. As orientações do capítulo anterior podem ser úteis para escolher a plataforma, mas talvez seja necessário verificar questões simples, mas que têm um enorme impacto na acessibilidade:

Como ficarão os títulos?

Essa é a primeira coisa que tecnologia assistiva para pessoas

cegas lê quando entra em uma página. Cerifique-se de que os títulos serão claros e que seguirão a hierarquia das páginas.

> **Quem são os beneficiados?** Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela) e com deficiência ou limitações cognitivas.

O idioma da página está definido?

É comum que plataformas definam um idioma padrão para a aplicação. Verifique se o idioma definido é o mesmo do conteúdo que será publicado.

> **Quem são os beneficiados?** Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela). Isso também ajuda as pessoas com deficiências ou limitações cognitivas ou de aprendizagem a se localizarem melhor no texto quando o conteúdo está estruturado de forma adequada.

A estrutura semântica está mantida?

Verifique se a plataforma oferece estrutura semântica (utilizando os principais elementos do HTML5, como `<header>`, `<section>`, `<article>` etc.). Se não estiver, verifique a possibilidade de alterar a estrutura com mais significado.

Quem são os beneficiados? Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela) e pessoas com deficiência intelectual ou limitações cognitivas.

Existem cabeçalhos na página? Estão estruturados?

A estrutura de cabeçalhos faz a marcação de áreas importantes do site. Tenha em mente que esses cabeçalhos devem estar estruturados de forma que o usuário consiga entender sua estrutura. Verifique também a hierarquia dos cabeçalhos que serão adicionados por publicadores de conteúdo na aplicação.

Quem são os beneficiados? Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela), com deficiência intelectual ou limitações cognitivas e usuários com limitações motoras.

Existem formas para saltar conteúdo repetido?

Além da estrutura semântica, que já permite o salto para áreas como formulários e cabeçalhos, verifique se há uma forma de saltar conteúdo repetido na página entre os primeiros links. Se não houver, verifique se há uma forma de adicionar esse link relacionado ao conteúdo principal da página/aplicação.

Quem são os beneficiados? Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela) e usuários com limitações motoras.

Os componentes são acessíveis?

Antes de adicionar componentes desenvolvidos por terceiros verifique se eles são acessíveis. Calendários, carrosséis e players devem ser testados para evitar barreiras de acesso.

Quem são os beneficiados? Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela), com deficiência intelectual ou limitações cognitivas e usuários com limitações motoras.

É possível adicionar atributos de WAI-ARIA?

Em áreas estruturais talvez não seja necessário devido à semântica dos elementos, mas em widgets e áreas dinâmicas talvez seja necessário adicionar atributos de `WAI-ARIA` caso o componente não seja acessível.

Quem são os beneficiados? Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela)

Os formulários estão acessíveis?

Independente de seu formulário ter sido criado manualmente ou pela plataforma escolhida, verifique se os rótulos estão relacionados adequadamente e se o formulário não tem barreiras para a navegação por teclado.

Quem são os beneficiados? Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela), com deficiência intelectual ou limitações cognitivas e usuários com limitações motoras.

Existem barreiras para a navegação por teclado?

Não só os formulários, mas todos os elementos interativos devem permanecer acessíveis por teclado. Verifique se a plataforma tem alguma barreira ou bloqueio na navegação quando o usuário não usa o mouse.

Quem são os beneficiados? Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela), com deficiência intelectual ou limitações cognitivas e usuários com limitações motoras.

Conheça bem os padrões Web!

Antes de manipular suas bibliotecas e frameworks preferidos é fundamental ter um bom conhecimento dos principais padrões Web para a construção de aplicações. HTML, CSS e JavaScript são a base da Web e amplamente utilizados por qualquer plataforma. Sem o conhecimento desses padrões fica muito complicado identificar eventuais problemas de acessibilidade no desenvolvimento da sua aplicação.

10.3 PUBLICAÇÃO DE CONTEÚDO

Finalmente, depois de organizar toda a estrutura, chegou a hora de publicar conteúdo. A publicação de conteúdo consiste em duas etapas: a primeira, do conteúdo estrutural da página; e a segunda, do que é publicado por interfaces administrativas. Nem sempre é possível customizar a publicação das áreas administrativas, ou isso requer conhecimentos de HTML por jornalistas ou publicadores de conteúdo para tanto, mas com pequenas intervenções é possível contemplar a acessibilidade ao subir conteúdo em um site.

Os cabeçalhos estão estruturados?

Com base na ordem de cabeçalhos da aplicação (por exemplo, se o `<h1>` está no logo, no título da notícia ou no nome da seção da aplicação) verifique como vai dar continuidade à ordem dos cabeçalhos. Conheça a estrutura e publique conteúdo com base

nela. Por exemplo, em uma matéria sobre o time do Palmeiras (definido como `<h2>`), o título da matéria deve ser um `<h3>` e uma nota sobre a escalação do time pode ter um título `<h4>`.

> **Quem são os beneficiados?** Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela) e usuários com limitações motoras

As imagens têm texto alternativo?

Todas as imagens que transmitem informação e fazem parte do contexto da página devem ter uma alternativa em texto. As imagens estruturais devem ter seus textos alternativos definidos lá no planejamento, mas figuras publicadas na aplicação por interfaces administrativas também precisam contemplar esse recurso. Na maioria dos CMSs existe um campo para texto alternativo da imagem e uma forma de editar o código-fonte do post ou página.

Se não existir uma forma de adicionar o texto alternativo, entre na interface de visualização do código do post e procure por `<img` e verifique se existe um atributo `alt=""`. Se não existir, adicione este atributo e descreva a imagem entre as aspas.

Tenha cuidado também com o conteúdo do atributo `title` na imagem. Muitas vezes ele repete o conteúdo do atributo `alt`. Para imagens, o atributo `title` serve para adicionar informação complementar. Ele não é obrigatório, sendo assim, se não houver necessidade de uso você pode descartá-lo.

Quem são os beneficiados? Pessoas cegas e com baixa visão (usuários de leitores ou ampliadores de tela) e com deficiência intelectual ou limitações cognitivas.

Os textos estão compreensíveis?

Quando digo sobre textos compreensíveis me refiro a diversas situações, como frases e parágrafos não muito longos, evitar texto justificado e figuras de linguagem. Pessoas com limitações cognitivas podem ter dificuldade ao ler e interpretar esse tipo de texto.

Quem são os beneficiados? Pessoas com deficiências ou limitações cognitivas, neurológicas ou de aprendizagem; pessoas com baixo letramento.

Os vídeos têm legendas?

Não esqueça de adicionar legendas aos vídeos publicados!

Quem são os beneficiados? Pessoas surdas e com baixa audição ou que não podem ouvir o conteúdo naquele momento devido a uma limitação situacional.

O CONTEÚDO É FEITO PARA O USUÁRIO!

De nada adianta publicar um ótimo conteúdo em um site se o usuário não consegue ler, escutar ou compreender. Publicar algo na rede significa entregar para a audiência o seu produto. Pensar na acessibilidade possibilita não só ampliar o público que vai consumir esse conteúdo, mas novas formas de acesso. Um conteúdo bom e acessível mantém seu público fiel.

10.4 PRÓXIMOS PASSOS

A maioria das dicas deste capítulo podem ser adicionadas à sua rotina sem aumento tempo de desenvolvimento ou instalação de plugins na página. São práticas simples que melhoram muito a acessibilidade das páginas desenvolvidas. Quando essas dicas entram na rotina de desenvolvimento, o processo para acompanhar a acessibilidade fica muito mais simples.

Agora que você já tem ótimas dicas para colocar na rotina do desenvolvimento, que tal entender como verificar e auditar a acessibilidade? Esse é o tema que abordaremos no próximo capítulo. Verificação de acessibilidade é um processo complementar ao de planejar o desenvolvimento acessível desde o início.

Capítulo 11

VERIFICAÇÃO DE ACESSIBILIDADE

O processo de verificação de acessibilidade começa durante o processo de desenvolvimento. Quando seguimos as recomendações dos capítulos anteriores já estamos colocando em nosso código boas práticas de acessibilidade e eliminando boa parte das barreiras de acesso. Esse processo contínuo é fundamental para evitar surpresas em uma eventual auditoria de acessibilidade.

Mas existem situações, principalmente em sites antigos e sistemas legados, em que a acessibilidade não foi considerada e é necessário fazer o diagnóstico da situação. Nesse caso, a verificação pós-desenvolvimento acaba sendo utilizada.

O objetivo deste capítulo é apresentar algumas formas de verificação dos principais pontos da acessibilidade de uma página, fazendo uso tanto de uma verificação com ferramentas automáticas quanto de testes manuais. A combinação dessas duas técnicas possibilita identificar o cenário da acessibilidade real.

Chamo de acessibilidade real a forma efetiva de eliminação de barreiras. É muito comum ver testes de acessibilidade em busca apenas a pontuação do validador, buscando de 100% de *conformidade* sem fazer testes manuais. Testes manuais podem identificar barreiras que os verificadores automáticos não encontraram.

11.1 VERIFICAÇÃO AUTOMÁTICA DE ACESSIBILIDADE

Consiste no uso de ferramentas automatizadas que fazem testes nas páginas Web com base em padrões e diretrizes de acessibilidade. Essas ferramentas acessam o código HTML e analisam a página. O resultado é entregue ao usuário na forma de erros e avisos.

Antes de fazer testes com validadores de acessibilidade, verifique se o *markup* está de acordo com os padrões do W3C. Utilize o verificador de HTML do próprio W3C (https://validator.w3.org/) para checar se o código que foi desenvolvido não tem erros ou avisos que podem causar problemas tanto de exibição como eventuais barreiras de acessibilidade.

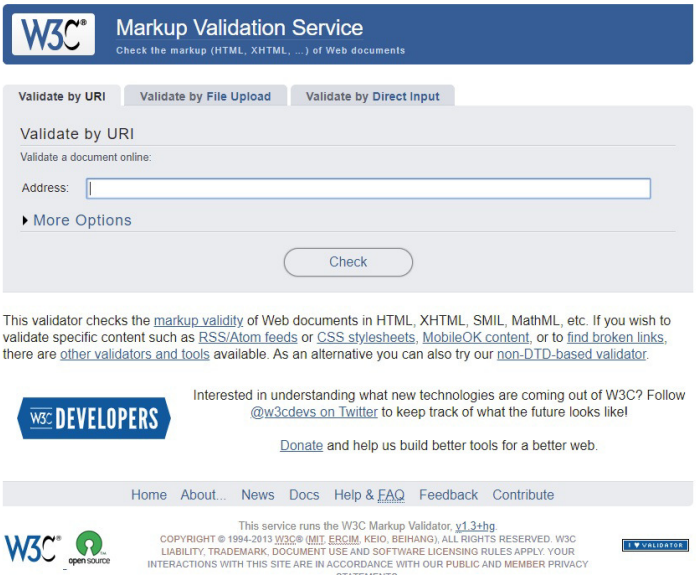

Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents
Validate by URI
Validate by File Upload
Validate by Direct Input
Validate by URI
Validate a document online:
Address:
More Options
Check
This validator checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS stylesheets, MobileOK content, or to find broken links, there are other validators and tools available. As an alternative you can also try our non-DTD-based validator.
DEVELOPERS
Interested in understanding what new technologies are coming out of W3C? Follow @w3cdevs on Twitter to keep track of what the future looks like!
Donate and help us build better tools for a better web.
Home About... News Docs Help & FAQ Feedback Contribute
open source
This service runs the W3C Markup Validator, v1.3+hg.
COPYRIGHT © 1994-2013 W3C® (MIT, ERCIM, KEIO, BEIHANG). ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY

Figura 11.1: Imagem da tela inicial do verificador de markup do W3C

O validador de markup do W3C também pode ser instalado em seu servidor, sem a necessidade de enviar solicitações frequentes aos servidores do W3C. O link (https://validator.w3.org/docs/install.html) tem informações detalhadas para a instalação local do validador.

O W3C também disponibiliza outros tipos de verificadores automáticos, como Feed RSS (https://validator.w3.org/feed/), CSS (http://jigsaw.w3.org/css-validator/) e até um verificador de links quebrados (https://validator.w3.org/checklink).

Feita a verificação de *markup*, passamos para os verificadores automáticos de acessibilidade.

Existe uma grande variedade de verificadores automáticos de

acessibilidade. Somente na página de Ferramentas de Verificação de Acessibilidade da WAI (https://www.w3.org/WAI/ER/tools/) eram 129 até a publicação deste livro.

Antes de apresentar algumas ferramentas e como utilizá-las, é importante lembrar que o próprio W3C não endossa as ferramentas da lista. Isso significa que obter 100% de conformidade com essas ferramentas não significa que esteja de acordo com seus padrões e nem garante sua acessibilidade. O *disclaimer* do W3C diz o seguinte (em uma tradução livre):

> **Isenção de responsabilidade importante**
>
> O W3C não endossa produtos de fornecedores específicos. A inclusão de produtos nesta lista não indica endosso pelo W3C. Os produtos e os critérios de pesquisa são listados sem classificação de qualidade.
> Descrições de ferramentas, critérios de pesquisa e outras informações neste banco de dados são fornecidos por desenvolvedores de ferramentas, fornecedores ou outros. O W3C não verifica a precisão das informações.
> A lista não é uma revisão das ferramentas de avaliação, nem uma lista completa ou definitiva de todas as ferramentas. As informações podem mudar a qualquer momento.

Tendo em mente que uma ferramenta automática não garante a acessibilidade sozinha, mas é uma grande aliada no processo de identificação de barreiras, considere os seguintes passos para a escolha da ferramenta:

- Em conformidade com WCAG 2.1 ou 2.0: apesar da grande lista de ferramentas, algumas delas ainda estão em conformidade apenas com WCAG 1.0. Utilizar os padrões mais modernos garante uma melhor identificação das barreiras.
- Escolha mais de uma ferramenta: as ferramentas costumam apresentar resultados diferentes entre si, devido à metodologia de análise dos critérios do WCAG. Tenha em mente que esses erros distintos podem ser bons para detectar mais barreiras de forma automática.

Vou apresentar como alguns dos validadores se comportam e como fazer uso das informações que eles apresentam. Apesar de detalhar o processo de cada um deles, também não estou dizendo que devam ser utilizados especificamente esses validadores ou somente esses. Fica a critério do usuário escolher a ferramenta que trará os resultados de que necessita.

Aqui veremos sobre os seguintes validadores:

- WAVE - http://wave.webaim.org/
- HTML_Codesniffer - http://squizlabs.github.io/HTML_CodeSniffer/
- AChecker - https://achecker.ca/checker/
- MAUVE - https://mauve.isti.cnr.it/
- AsesWeb - http://asesweb.governoeletronico.gov.br/ases/
- Access Monitor Plus - http://accessmonitor.acessibilidade.gov.pt/amp/

Para verificá-los, exibirei alguns testes dessas ferramentas em alguns sites, mas sem identificar o site avaliado.

WAVE

Ferramenta criada pela WebAim, uma empresa que atua com acessibilidade na Web desde 1999. Também baseada em WCAG 2.0, a ferramenta tem uma interface inicial simples, apenas com um campo de formulário para inserir a URL da página. Diferente do Access Monitor, não verifica por upload de arquivo ou código-fonte da página.

A falta de opções na página inicial é compensada pela quantidade de informações exibidas na interface com o resultado da pesquisa. Do lado esquerdo da tela está o menu de navegação e os resultados da verificação. Do lado direito está a página carregada com marcações onde existem erros a serem corrigidos.

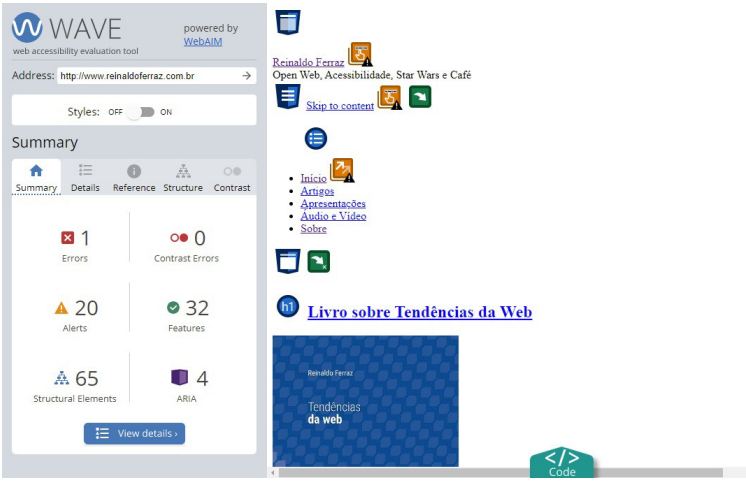


Figura 11.2: Imagem com o resultado da ferramenta WAVE avaliando um site

O menu vertical permite a exibição da página com e sem os estilos (CSS) para uma melhor visualização dos resultados. Logo abaixo o usuário encontra um resumo da verificação com os

detalhes encontrados, como o número de erros, alertas, erros de contraste além de mostrar a estrutura do documento, recursos e atributos de WAI-ARIA.

Ela tem um recurso muito bom embutido que é a verificação do contraste entre o texto e o fundo da página seguindo as diretrizes do WCAG.

Apesar disso, a ferramenta não consegue identificar as imagens de fundo. Dessa forma, ela vai avaliar uma fonte sobre uma imagem com base na cor de fundo e não na imagem. Isso pode gerar um "falso positivo", se a imagem proporcionar um contraste adequado com o texto.

O menu vertical abaixo da barra de endereço de pesquisa do site permite navegar pelos resultados da avaliação. O primeiro ícone apresenta um resumo da avaliação. O segundo mostra os detalhes da verificação. Ele identifica cada erro e aviso e permite uma análise detalhada sobre eles. Ao clicar em cada item ele mostra o detalhe no box ao lado. O botão para a referência serve como uma base de dados para os ícones apresentados na avaliação e seu respectivo significado. O botão seguinte apresenta a estrutura da página, com os elementos utilizados e o último botão mostra o contraste entre a cor do texto e sua respectiva cor de fundo no site.

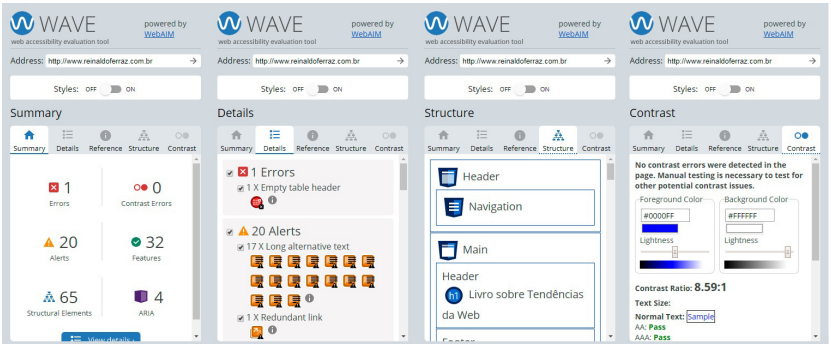


Figura 11.3: Imagem exibindo o conteúdo de cada item do menu vertical da ferramenta WAVE

Já na área central/direita da aplicação é exibida a página renderizada com as marcações do validador. As marcas relacionadas a erros permitem acessar mais informações sobre o problema encontrado. Dessa forma fica mais fácil compreender o erro sem perder a visualização da página.

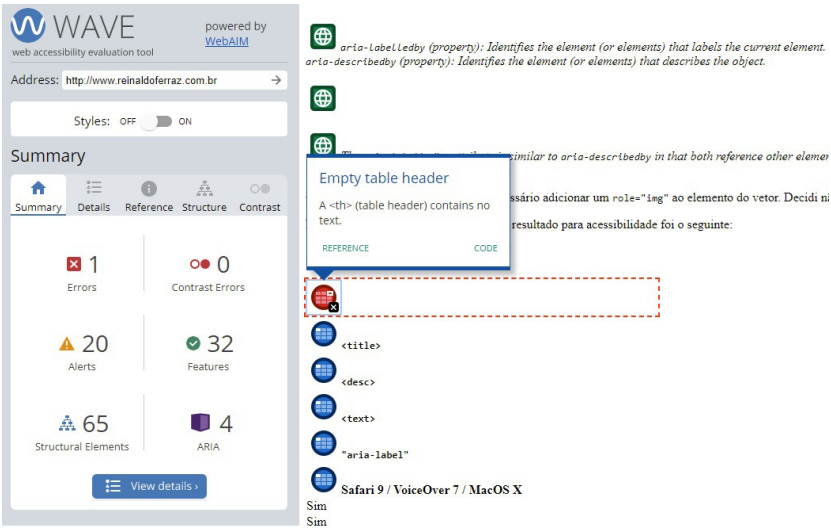


Figura 11.4: Imagem do resultado da verificação da ferramenta WAVE mostrando um box ao clicar no erro. O detalhamento do erro é exibido no painel à esquerda.

A WebAim permite a avaliação de sites completos em sua ferramenta Dinolytics, que requer a contratação de um plano pago.

Pontos positivos:

- Suporte a WCAG 2.0
- Verificação de contraste entre fundo e texto
- Permite exibição da página com e sem estilos
- Interface bem visual e intuitiva

Pontos negativos:

- Contraste não detecta imagens de fundo
- Não permite verificação por upload de arquivo ou código direto
- Não suporta ainda WCAG 2.1

11.2 HTML_CODESNIFFER

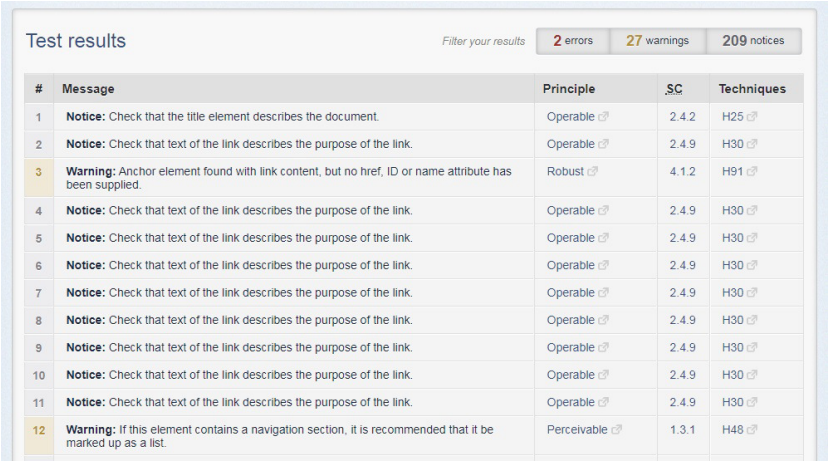
Desenvolvido pela empresa Squiz, uma empresa australiana de produtos digitais, o verificador HTML_Codesniffer tem uma interface simples, mas bastante eficiente, para a exibição dos resultados de verificação.

Basicamente o que você precisa fazer é copiar o código fonte da sua página e colar no campo texto na parte inferior da página inicial. Esse campo poderia ficar no topo da página, mas eles têm uma boa razão para isso (que explico mais adiante). A ferramenta verifica a conformidade com WCAG 2.0 nos níveis A, AA e AAA (além de conformidade com Section 508). Não faz verificação por URL em sua aplicação Web.

HTML_CodeSniffer
Squiz
github
Check that your HTML code conforms to your coding standard
HTML_CodeSniffer is a client-side script that checks HTML source code and detects violations of a defined coding standard. HTML_CodeSniffer is written entirely in JavaScript, does not require any server-side processing and can be extended by developers to enforce custom coding standards by creating your own "sniffs".
To get you started, HTML_CodeSniffer comes with standards that enforce the three conformance levels of the Web Content Accessibility Guidelines (WCAG) 2.0, and the web-related components of the U.S. "Section 508" legislation. An auditor interface is provided by a bookmarklet to let you try out these accessibility checks on any web page.
List of standards and sniff coverage
WCAG 2.0 (Level A, AA, AAA)
Section 508
HTML_CODESNIFFER BY SQUIZ
Standards:
WCAG2AA
1
Error
5
Warnings
14
Notices
Select the types of issues to include in the report
View Report
Issues detected by the bookmarklet:
79,385,117
Get the Accessibility Auditor Bookmarklet
Drag this link to your bookmarks bar to install
Auditor version: 2.1.1
This bookmarklet has been designed to work with Google Chrome, Firefox, Safari and Internet Explorer
Run your code through the Sniffer
Run HTML_CodeSniffer over your own code to see how it works. The sniffer below will test your source code against your selected guideline.
Test results will be shown underneath the code block.
WCAG 2.0 Level A
WCAG 2.0 Level AA
WCAG 2.0 Level AAA
Section 508
Run HTML_CodeSniffer
HTML_CodeSniffer developed by Squiz Content Management Solutions

Figura 11.5: Imagem da página inicial da ferramenta HTML_Codesniffer que tem um campo texto para inserir o código-fonte da página cuja acessibilidade se deseja verificar.

O resultado é exibido em formato de tabela, de forma simples e com links para as diretrizes e critérios de sucesso do WCAG relacionados. É possível exibir somente os erros (errors), avisos (warnings) e observações (notices). Este último é bem interessante, pois entre as observações verifica-se se o conteúdo de atributos complementares (como `title`) descreve claramente o objeto (link, imagem etc.).



Test results

Filter your results | 2 errors | 27 warnings | 209 notices

#	Message	Principle	SC	Techniques
1	**Notice:** Check that the title element describes the document.	Operable	2.4.2	H25
2	**Notice:** Check that text of the link describes the purpose of the link.	Operable	2.4.9	H30
3	**Warning:** Anchor element found with link content, but no href, ID or name attribute has been supplied.	Robust	4.1.2	H91
4	**Notice:** Check that text of the link describes the purpose of the link.	Operable	2.4.9	H30
5	**Notice:** Check that text of the link describes the purpose of the link.	Operable	2.4.9	H30
6	**Notice:** Check that text of the link describes the purpose of the link.	Operable	2.4.9	H30
7	**Notice:** Check that text of the link describes the purpose of the link.	Operable	2.4.9	H30
8	**Notice:** Check that text of the link describes the purpose of the link.	Operable	2.4.9	H30
9	**Notice:** Check that text of the link describes the purpose of the link.	Operable	2.4.9	H30
10	**Notice:** Check that text of the link describes the purpose of the link.	Operable	2.4.9	H30
11	**Notice:** Check that text of the link describes the purpose of the link.	Operable	2.4.9	H30
12	**Warning:** If this element contains a navigation section, it is recommended that it be marked up as a list.	Perceivable	1.3.1	H48

Figura 11.6: Imagem do resultado de verificação de acessibilidade da ferramenta HTML_Codesniffer.

A aplicação não mostra em que linha ou parte do site está o erro. Sendo assim, ela serve apenas para uma verificação prévia da acessibilidade da página. Mas a parte mais robusta e completa da aplicação vem a seguir.

Na página inicial existe um banner do lado direito da tela. Esse banner exibe a seguinte mensagem:

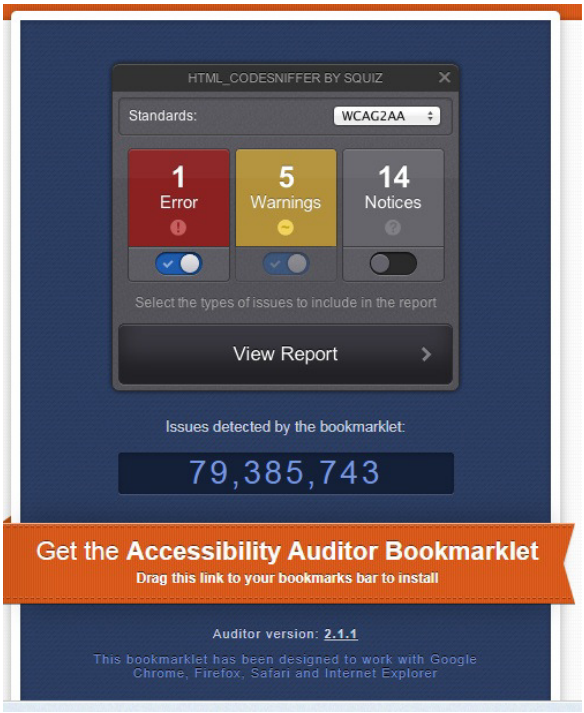


Figura 11.7: Imagem do banner na página inicial da ferramenta HTML_Codesniffer.

Esse banner apresenta o "Accessibility Auditor Bookmarklet" e orienta o usuário a "arrastar" este banner para os seus favoritos para ativar a ferramenta. O link nada mais é do que um JavaScript que detecta a página na barra de endereços e exibe o auditor de HTML da página (disponível apenas para Google Chrome, Firefox, Safari e Internet Explorer).

Quando aplicada a barra de favoritos do navegador (e clicada) ela mostra o seguinte resultado:

Reinaldo Ferraz
Open Web, Acessibilidade, Star Wars e Café
INÍCIO
ARTIGOS
APRESENTAÇÕES
ÁUDIO E VÍDEO
SOBRE
Livro sobre Tendências da Web
Reinaldo Ferraz
Tendências
da web
O segundo livro que escrevi para a Pós-Graduação em Tecnologias e Inovações para Web já está disponível em formato digital para compra nas lojas Amazon, Kobo e no Google Books.
Esse foi o resultado de seis meses de pesquisa e dedicação. Junto com o
HTML_CODESNIFFER BY SQUIZ
Standards:
WCAG2A
5
Errors
18
Warnings
191
Notices
Select the types of issues to include in the report
View Report
RT @cewebbr: Fim do 1º lote! Último dia para garantir ingressos para a #Webbr2019 no valor promocional. A Conferência reunirá es...
https://t.co/FZxMEnSLIm
6 meses ago

Figura 11.8: Imagem exibindo o box de avaliação da ferramenta HTML_Codesniffer no site de Reinaldo Ferraz.

Com essa ferramenta é possível saber em que parte do código está o problema. Ao clicar em "View Report" ela lista os erros, avisos e observações. Ao clicar em algum deles um ícone aponta onde está o erro na interface visual do código e o código do elemento com problemas é exibido no box.

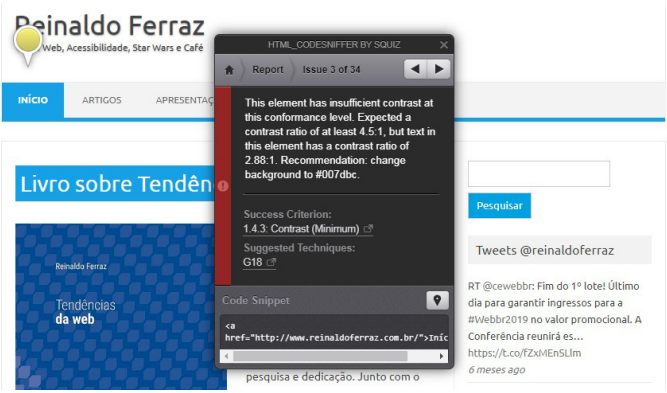

Reinaldo Ferraz
Web, Acessibilidade, Star Wars e Café
INÍCIO
ARTIGOS
Livro sobre Tendên
Reinaldo Ferraz
Tendências
da web
HTML_CODESNIFFER BY SQUIZ
Report
Issue 3 of 34
This element has insufficient contrast at this conformance level. Expected a contrast ratio of at least 4.5:1, but text in this element has a contrast ratio of 2.88:1. Recommendation: change background to #007dbc.
Success Criterion:
1.4.3: Contrast (Minimum)
Suggested Techniques:
G18
Code Snippet
<a href="http://www.reinaldoferraz.com.br/">Iníc
pesquisa e dedicação. Junto com o
Pesquisar
Tweets @reinaldoferraz
RT @cewebbr: Fim do 1º lote! Último dia para garantir ingressos para a #Webbr2019 no valor promocional. A Conferência reunirá es...
https://t.co/FZxMEnSLIm
6 meses ago

Figura 11.9: Imagem exibindo o box de avaliação detalhado da ferramenta HTML_Codesniffer no site de Reinaldo Ferraz.

Apesar dos benefícios, a ferramenta não permite a verificação de uma página do seu computador exibida no navegador. Toda página a ser verificada precisa estar em um endereço na Web. Para a verificação de código local é necessário inserir o código copiado em sua interface Web.

Pontos positivos:

- Suporte a WCAG 2.0
- Ferramenta simples e intuitiva
- Sugere observações além da documentação do WCAG
- Aplicação para verificação rápida de qualquer site

Pontos negativos:

- Não existe versão em português
- Permite somente verificação de páginas disponíveis na Web (não permite verificação local)
- Não suporta ainda WCAG 2.1

11.3 ACHECKER

Ferramenta desenvolvida em 2009 pelo Centro de Pesquisa em Design Inclusivo da Universidade de Toronto. É uma ferramenta simples para verificar a acessibilidade de uma página Web.

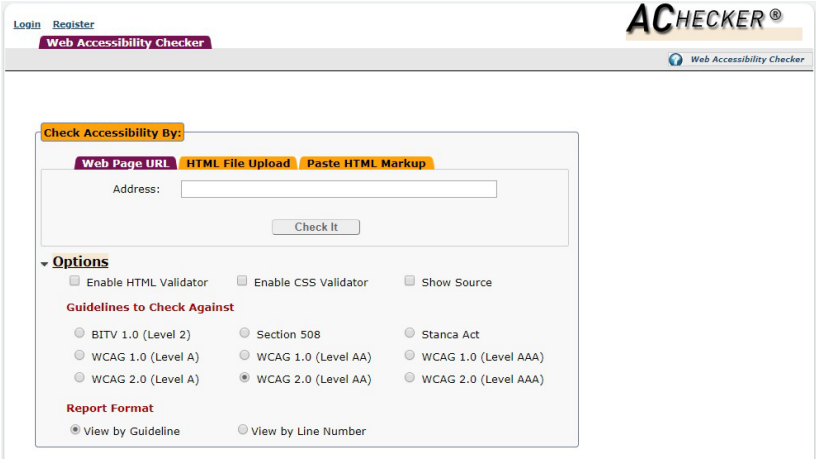


Figura 11.10: Imagem da página inicial do validador AChecker

Permite a verificação de página por URL, por código-fonte direto ou upload de arquivo. Isso permite não só a verificação de URLs da rede como o de páginas locais, como arquivos do seu próprio computador.

Logo abaixo da barra de pesquisa existe um link (options) que expande as opções de pesquisa: Verificação de markup do HTML, verificação de CSS, exibição de código-fonte além da conformidade por BITV 1.0, Section 508, Stanca act e WCAG 1.0 e 2.0 (níveis A, AA e AAA).

O resultado é bem completo e detalhado. Ele aponta exatamente onde está o erro no código e mostra algumas linhas do código com problemas. Muito útil para detectar o erro diretamente no código.

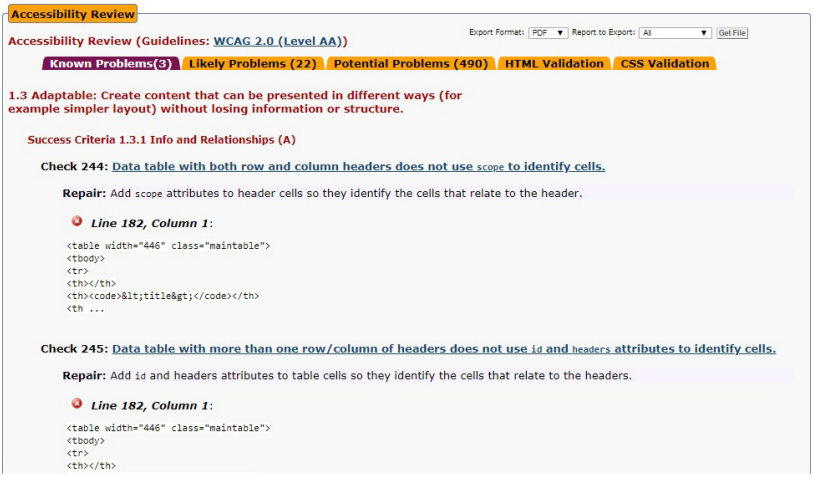


Figura 11.11: Imagem da página de resultados da verificação do validador AChecker

Os resultados são separados por categorias: Problemas conhecidos (known problems), problemas prováveis (likely problems), problemas em potencias (potential problems) e verificação de HTML e CSS.

Resumindo: a aplicação mostra onde existem problemas e onde podem existir. Problemas em potencial são situações que podem vir a trazer problemas para o usuário, mas que não são necessariamente erros, por exemplo, imagens com texto alternativo longo.

Além disso é possível exportar o relatório em HTML, CSV e PDF.

Pontos positivos:

- Suporte a WCAG 2.0
- Detalha exatamente onde estão os erros do código

- Possibilidade de verificação de URL, código direto ou upload de arquivo
- Apresenta "potenciais problemas" no código
- Exporta os relatórios

Pontos negativos:

- Não existe versão em português
- Interface poluída
- Não suporta ainda WCAG 2.1

MAUVE

Verificador de acessibilidade desenvolvido pelo Laboratório de interfaces humanas em sistemas de informação do Instituto de Ciência da Informação e Tecnologia do Conselho Nacional de Pesquisa da Itália. Tem uma interface inicial um pouco mais complexa que os anteriores, mas essa complexidade traz recursos importantes para a verificação da acessibilidade.

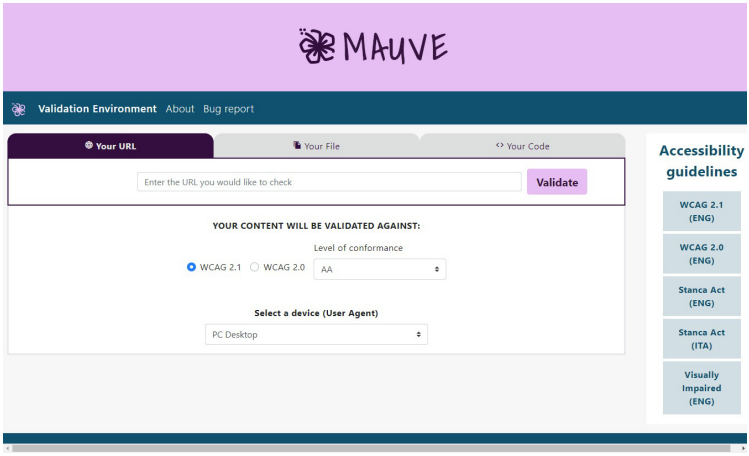


Figura 11.12: Imagem da página inicial do validador MAVUE

Na página inicial estão disponíveis diversas opções para a verificação de acessibilidade. Você pode escolher verificar uma página por uma URL, por upload de código ou por entrada direta do código na ferramenta. Em seguida, você pode optar por verificação com base no padrão WCAG 2.1 ou WCAG 2.0 e escolher o nível de conformidade (entra A, AA e AAA).

O campo seguinte permite selecionar a verificação de acessibilidade com base no tipo de dispositivo. Nos testes feitos em diversos dispositivos e com vários tipos de aplicações (sites e apps/PWA) não encontrei diferença na verificação, já que o resultado da verificação foi sempre para o agente de usuário computador/desktop.

O resultado da verificação apresenta um breve resumo do que foi verificado e encontrado, mas os detalhes que ele apresenta são muito valiosos.

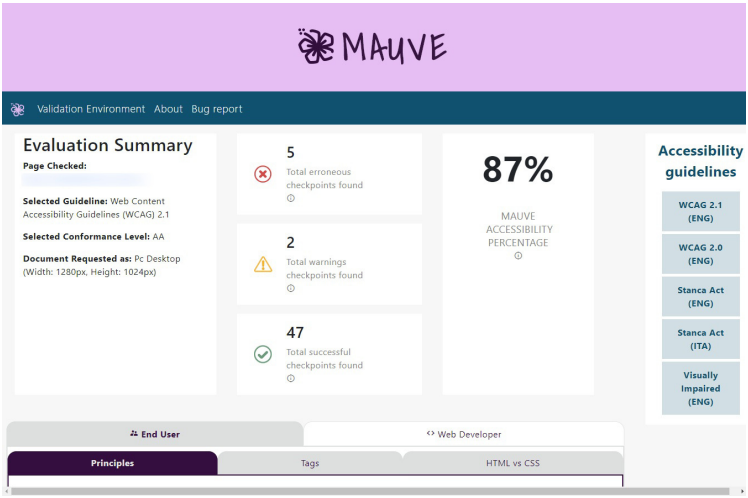


Figura 11.13: Imagem do resultado do validador MAVUE

Além de mostrar a quantidade de erros, avisos e itens em conformidade no resumo, abaixo o validador apresenta os detalhes do que foi verificado. São cinco abas que apresentam os seguintes resultados:

- ***End user*** (Usuário final): lista de forma simples e intuitiva os tipos de erro, sem detalhar ou remeter a códigos, de forma a facilitar a compreensão de um usuário sem conhecimento técnico.
    - ***Principles***: apresenta os princípios do WCAG relacionado aos erros e avisos encontrados.
    - ***Tags***: uma das mais interessantes abas, pois apresenta um relatório com os diversos erros relacionados ao tipo de objeto da página, como WAI-ARIA, formulários, HTML, imagens, tipografia e até erros relacionados à responsividade da aplicação.
    - ***HTML vs CSS***: separa os resultados entre os erros relacionados à marcação (HTML) e apresentação (CSS)
- ***Web Developer*** (Desenvolvedor): exibe o código-fonte da página apresentando os erros e os relacionando com os critérios do WCAG 2.1/2.0. Apesar de apontar diretamente o link para a documentação do W3C, ele não detalha muito o que deve ser feito para corrigir o erro. Também não lista os erros para identificar no código, exigindo que o usuário passe por todo o código para encontrar os erros marcados.

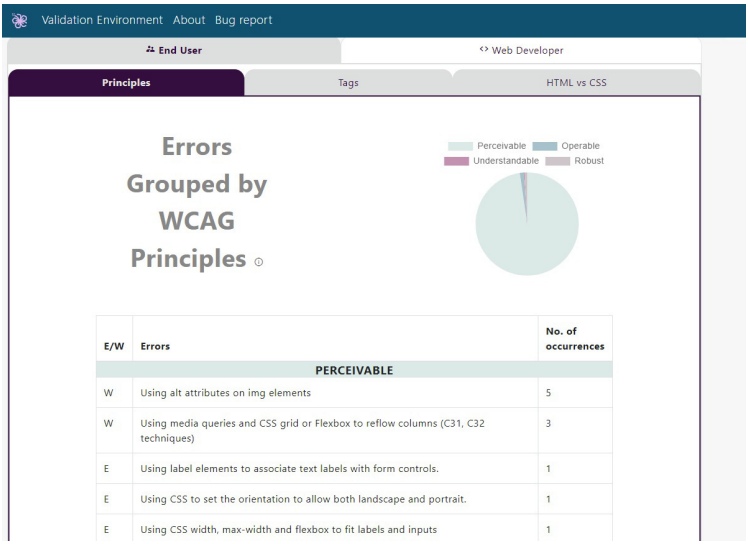


Figura 11.14: Imagem do resultado para detalhamento de erros para o usuário final do validador MAVUE

Validation Environment
About
Bug report
SC 1.4.12 - Tech C8 [WCAG 2.1 (AA)]Using CSS letter-spacing to control spacing within a word
172
SC 1.4.12 - Tech C8 [WCAG 2.1 (AA)]Using CSS letter-spacing to control spacing within a word
173
SC 2.5.3 - Tech F96 [WCAG 2.1 (A)]If on the page are present the tags aria-label or aria-labelledby, check that their content matches the visible name.
SC 1.3.1 - Tech G140 [WCAG 2.1 (A)]Separating information and structure from presentation to enable different presentations
174
175 <svg>
176 <use xlink:href="#HU-arrow-down"></use>
177 </svg>
178
179
180
181
182
SC 1.4.12 - Tech C8 [WCAG 2.1 (AA)]Using CSS letter-spacing to control spacing within a word
183
SC 1.4.12 - Tech C8 [WCAG 2.1 (AA)]Using CSS letter-spacing to control spacing within a word

Figura 11.15: Imagem do resultado para detalhamento de erros diretamente no código do validador MAVUE

Pontos positivos:

- Suporte a WCAG 2.1 e 2.0
- Verificação de erros relacionados a design responsivo
- Interface de relatório para usuário sem conhecimento técnico
- Permite identificação fácil entre erros de HTML e CSS
- Permite verificação por URL, upload de arquivo e código-fonte

Pontos negativos:

- A caixa de seleção de agente de usuário não faz diferença na verificação
- A inspeção de código não detalha a forma de solucionar o erro

Ases Web

É um validador desenvolvido pelo Governo Federal brasileiro com o objetivo de auxiliar na construção de páginas Web com base no eMag (Modelo de Acessibilidade do Governo Federal). Como já vimos nos capítulos anteriores, o eMag é baseado nas diretrizes do WCAG para que sites governamentais sejam acessíveis para pessoas com deficiência.

O Ases surgiu inicialmente como um software de plataforma feito em Java e evoluiu para uma interface Web. Porém, diversos recursos adicionais para verificação de acessibilidade estão disponíveis somente na versão instalável. Destaco entre essas ferramentas o simulador de baixa visão (permite simular como uma pessoa daltônica ou com miopia, catarata e retinopatia

enxergam a página) e a possibilidade de verificar múltiplas páginas de um website, esteja ele online ou em uma pasta remota. Ele pode ser baixado no portal do Software Público, em https://softwarepublico.gov.br/social/ases/.

Sua interface inicial segue o padrão das demais: simples e intuitiva.

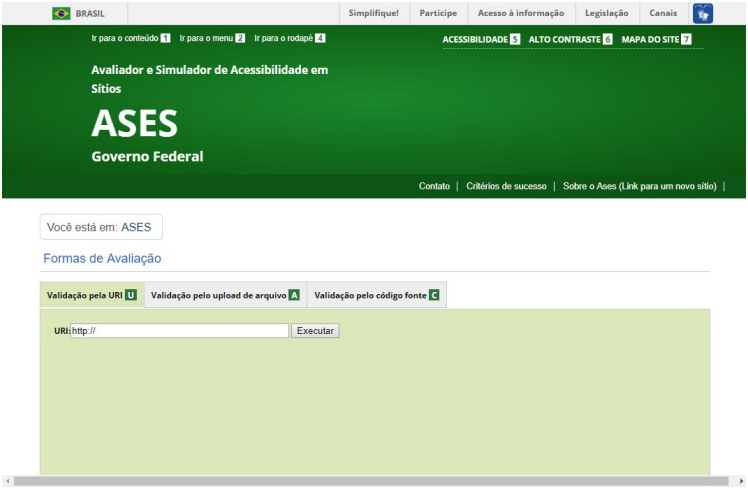


Figura 11.16: Imagem da tela inicial do validador Ases Web

Além da verificação por URL, o ASES permite a verificação por upload de arquivo e inserção direta de código.

Um dos pontos positivos do ASES é que a maior parte de toda informação sobre a verificação da página encontra-se na mesma página. No início da página há um breve resumo do que foi avaliado, o que gera uma nota baseado na conformidade com o padrão eMag, que vai de 0 a 100%. Esse resumo apresenta também os erros e avisos separados por tipos de erros com base no ASES. Esses tipos de erros podem ser comparados com os princípios do

WCAG, pois estão relacionados a marcação, comportamento, formulários etc.

Página Avaliada
Página:
Titulo:
Tamanho: 59642 Bytes
Data/Hora: 30/08/2019 07:37:03
Nota e Resumo da Avaliação de Acessibilidade
Porcentagem
ASES
87,47%
Legenda
>= 95%
>= 85% < 95%
>= 70% < 85%
< 70%
Resumo de Acessibilidade por Seção eMAG
Seção | Erro(s) | Aviso(s)
Marcação | 25 | 96
Comportamento | 1 | 4
Conteúdo/Informação | 20 | 16
Apresentação / Design | 0 | 0
Multimídia | 0 | 0
Formulários | 0 | 1
Total | 46 | 117
Avaliação tem por base testes automáticos em código-fonte (X)HTML interpretados do Modelo de Acessibilidade em Governo Eletrônico (eMAG) (link para novo sítio).
A nota não contempla os itens classificados como avisos e aqueles que requerem avaliação humana. Para saber quais testes são contemplados pelo software, favor verificar os critérios de sucesso trabalhados pelo ASESWEB.
Resumo de Acessibilidade por recomendações do eMAG
Marcação
Comportamento
Conteúdo/Informação
Apresentação / Design
Multimídia
Formulários
1.1 Respeitar os Padrões Web.(link para um novo sítio) 13 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1.2 Organizar o código HTML de forma lógica e semântica.(link para um novo sítio) 3 359, 415, 415
1.3 Utilizar corretamente os níveis de cabeçalho.(link para um novo sítio) 7 100, 125, 148, 288, 347, 454, 478

Figura 11.17: Imagem da tela do resultado de verificação do validador Ases Web

Mais abaixo nessa mesma página é exibida uma tabela com o detalhamento dos erros e avisos, que lista cada um deles com links para orientações detalhadas sobre como solucionar o erro. Há ainda a exibição de todo o código-fonte da página no resultado inicial, mas não há marcação dos erros nesse código.

O que me chamou a atenção no ASES é o detalhamento dos erros. Quando acessamos o link referente as linhas do código-fonte onde está o erro o ASES abre uma nova página mostrando exatamente as linhas onde estão.

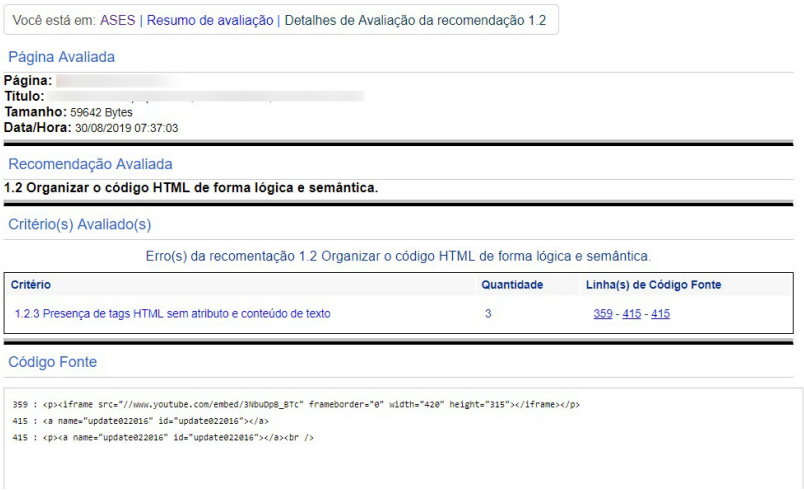


Figura 11.18: Imagem da tela do resultado detalhado da verificação do validador Ases Web

Pontos positivos:

- Totalmente em português do Brasil
- Interface de relatório de resultados amigável
- Permite exportar qualquer resultado de avaliação em PDF
- Permite verificação por URL, upload de arquivo e código-fonte
- Possui versão de avaliação de múltiplas páginas (no aplicativo instalável)

Pontos negativos:

- Apesar de o eMag ser baseado no WCAG, ele ainda não suporta as recomendações da versão 2.1

11.4 ACCESS MONITOR PLUS

Desenvolvido pela agência pública nacional para a ciência, tecnologia e inovação de Portugal (FCT), está baseado em WCAG 2.0. Ficou fora do ar no início de janeiro de 2020 e voltou em um novo endereço e nova identidade, apesar de poucas mudanças significativas. Tem uma interface que permite a verificação de URL de um site já publicado, mas também permite a análise por upload de arquivo HTML e inserção direta do código HTML. A ferramenta está em português, o que facilita a identificação das barreiras de acesso.

Você está em Acesso > Access Monitor Plus
AccessMonitor
o validador de práticas de acessibilidade Web
Access Monitor Plus
Obtenha um relatório de acessibilidade através de um dos seguintes métodos
Inserir URL
Inserir código HTML
Carregar ficheiro HTML
Inserir url aqui
Exemplo: http://www.google.pt
Avaliar

Figura 11.19: Imagem da tela inicial do verificador de acessibilidade

Depois de inserir a URL, arquivo ou código a ser verificado, o validador dá uma nota, com base nos seus critérios de avaliação do WCAG:

Figura 11.20: Imagem da tabela de resultado validador Access Monitor

Esse resultado apresenta informações importantes. Além da pontuação, que vai de zero a dez, é possível saber a quantidade de erros e avisos na página. Rolando para baixo na própria página de resultados a ferramenta traz informações detalhadas sobre eles.

Avaliação
Prática encontrada
Nível
Detalhes
Constatei que todas as imagens da página fazem uso do atributo alt. A
Encontrei 9 imagens na página em que alt tem mais de 100 carateres . A
Identifiquei 2 casos em que o atributo title do elemento link se limita a repetir o texto existente no link. A
Constatei que a primeira hiperligação da página não permite saltar diretamente para a área do conteúdo principal. A
Encontrei 1 links para contornar blocos de conteúdo. A
Encontrei 34 cabeçalhos na página. AAA

Figura 11.21: Imagem do detalhe dos resultados do validador Access Monitor exibindo um erro da página

Esse exemplo mostra como a ferramenta classifica os critérios de acessibilidade. Na imagem, ela exibe uma lista de critérios,

como texto alternativo em imagens, marcação de cabeçalhos, salto para conteúdo etc. Ao lado da notificação, uma pequena seta exibe o critério do WCAG 2.0 ao qual ele se refere. Nas colunas seguintes, é exibido o nível do erro (entre A, AA e AAA) e em algumas linhas há o ícone de uma "lupa", para detalhamento da verificação.

Essa lupa tem uma função interessante, pois mostra no código os elementos relacionados ao critério apontado na verificação. Quando o site tem imagens publicadas, por exemplo, é possível fazer a comparação entre o texto e a imagem. Ao acionar o ícone de "lupa" no critério de texto alternativo de imagem são exibidos todos os textos alternativos e as imagens exibidas, para fazer a comparação entre elas:

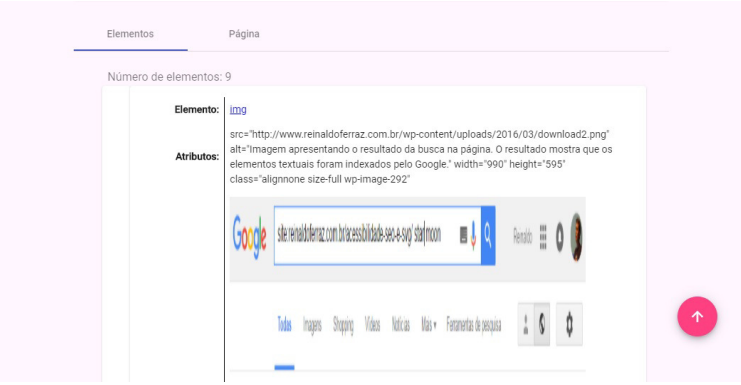


Figura 11.22: Imagem da comparação entre o texto alternativo da imagem e a figura exibida na página de resultados do Access Monitor

Isso serve não somente para imagens. Esse recurso permite detalhar cada elemento, como cabeçalhos, formulários etc.

Sobre a atualização da ferramenta - Verificada em março de 2020

A interface atual parece mais amigável, principalmente com relação aos textos do detalhamento do resultado. A combinação de cores está moderna e com um visual mais "clean". A ferramenta trata os textos como se alguém (talvez o robô que aparece próximo ao logo) esteja fazendo o reporte dos erros.
Ainda existem alguns problemas, principalmente de acessibilidade. Diversos textos importantes estão em cinza claro sobre fundo branco, o que pode dificultar a leitura dependendo do seu tamanho. Espero que esses pontos sejam corrigidos no futuro, pois é uma das poucas ferramentas de verificação em português.

Pontos positivos:

- Suporte a WCAG 2.0
- Boa interface para comparação de texto alternativo e imagens
- Suporte a múltiplos meios de verificação (URL/arquivo/código direto)
- Está em português

Pontos negativos:

- Verifica uma única página por vez
- Não suporta ainda WCAG 2.1

11.5 VERIFICAÇÃO MANUAL DE ACESSIBILIDADE

As ferramentas listadas e explicadas neste capítulo são ótimas para auxiliar no processo de identificação de erros de *markup* para a eliminação de barreiras de acesso, mas é fundamental fazer a verificação manual da página. Essa verificação permite que sejam detectadas barreiras que não podem ser verificadas (ainda) por ferramentas automáticas. Na maioria dos casos os alertas das ferramentas automatizadas trazem os principais pontos que devem ser verificados manualmente.

Vou listar alguns passos para auxiliar no processo de verificação manual de acessibilidade:

1 - Verifique se o site está acessível via teclado

Tente navegar por toda a sua aplicação sem a utilização do mouse ou trackpad. Tente acessar todos os itens interativos da página. Tenha atenção especial a:

- Menus e submenus
- Campos de formulário (principalmente os que atualizam campos automaticamente)
- Players de áudio e vídeo
- Links e botões

Verifique também se o foco do teclado está visível.

2 - Verifique o texto alternativo das imagens

Ferramentas automatizadas não conseguem (ainda) verificar se

o texto alternativo da imagem está de acordo com o que foi publicado. Certifique-se de que as imagens que fazem parte do contexto da página e aplicação têm seu atributo `alt` descrito de acordo com o que a imagem representa.

3 - Verifique o contraste de cores e tamanho de texto

Apesar de algumas ferramentas conseguirem identificar o contraste entre as cores de fundo e texto é necessário checar se o contraste está adequado. Algumas vezes são utilizadas imagens de fundo que a ferramenta cuja cor não é detectada pela ferramenta de verificação. Certifique-se ainda de que o zoom em até 200% no texto não quebra o design da página.

4 - Verifique se os vídeos e áudio tem alternativas em texto

Não é possível detectar textos em vídeos por ferramentas de verificação automática, sejam eles como legendas ou se estão apenas relacionados à transcrição por hiperlinks. Nesse caso, verifique se existem legendas (*open* ou *closed caption*) ou se existe transcrição do áudio nos vídeos.

No caso de conteúdos em áudio, como podcasts, verifique se existe transcrição do conteúdo em texto, seja ele na mesma página ou em um link específico para o conteúdo em texto.

5 - Verifique se há uma forma de parar conteúdo em movimento

É comum principalmente em banners ou carrosséis que o movimento se repita durante sua exibição na tela. Se existir algum

recurso que se move, verifique se existe alguma forma de parar ou pausar o conteúdo em movimento, seja utilizando um botão para parar/pausar ou pressionando alguma tecla.

6 - Verifique a hierarquia de cabeçalhos

Apesar de ser detectada por ferramentas automáticas de verificação, a organização de cabeçalhos deve contemplar uma ordem hierárquica de informações. Em alguns casos, essa ordem é feita com base na navegação, e em outros, como estrutura entre títulos e subtítulos. Não existe uma regra sobre o título de conteúdo nos cabeçalhos, desde que eles sigam uma ordem de hierarquia.

7 - Teste sua aplicação com tecnologia assistiva

Utilizar recursos de tecnologia assistiva pode ajudar a detectar barreiras específicas. Faça o uso de lupas e leitores de tela para verificar se é possível navegar pela página sem barreiras. Utilize a lista de tecnologia assistiva apresentada no capítulo 2 deste livro.

8 - Peça ajuda para pessoas com deficiência

Se for possível, peça auxílio a pessoas com deficiência para ajudarem a verificar a acessibilidade do seu site/aplicação. O resultado será muito rico e pode proporcionar uma nova forma de detectar barreiras de acessibilidade. Diversas instituições de apoio à pessoa com deficiência oferecem esse tipo de serviço.

11.6 MEU SITE ESTÁ DE ACORDO COM AS BOAS PRÁTICAS. E AGORA?

Quando chegamos nesse nível de definição talvez a primeira coisa que passa pela nossa cabeça seja coloca um banner na página espalhando para todos que seu site ou aplicação é acessível. Essa afirmação é delicada, pois a acessibilidade é viva, como o conteúdo de um site e aplicação, que sofre atualizações constantes. Se essa atualização não for feita com cuidado, a acessibilidade pode ser comprometida.

Acessibilidade não é binária. Seu site "está" ou "não está" acessível. Precisamos eliminar barreiras de acesso e definir os níveis de conformidade, assim eliminamos essas barreiras das páginas e promovemos a real inclusão de forma adequada.

Para tanto, vamos detalhar como deixar sua página em conformidade com padrões de acessibilidade e definir as barreiras de acesso a serem eliminadas, com base na documentação do W3C.

Capítulo 12

CONFORMIDADE E QUESTÕES LEGAIS

Acessibilidade na Web muitas vezes é invisível. Ela pode passar despercebida por uma pessoa que não tem deficiência ou por alguém que não tem experiência em avaliar barreiras de acesso. E isso pode ser delicado, pois a falta de detecção automática pode causar desconforto para os desavisados (que não veem um selo ou uma barra de acessibilidade) ou por entidades legais que desconhecem os critérios de avaliação.

Mas sites que estão acessíveis também podem ter barreiras de acesso, devido a uma atualização onde um requisito de acessibilidade pode ter sido esquecido.

Para evitar essas situações, este capítulo vai trazer alguns recursos para "informar aos desavisados" como foi feita a acessibilidade do seu site e aplicação.

12.1 DECLARAÇÃO DE CONFORMIDADE

Como já vimos nos capítulos anteriores, apenas um validador de acessibilidade não garante que um site seja plenamente acessível. Por isso, existem as "declarações de conformidade" para

declarar que sua aplicação está de acordo com as boas práticas e padrões adotados internacionalmente.

Essa declaração costuma ser publicada principalmente em sites no exterior. Nos Estados Unidos, por exemplo, os sites deve estar de acordo com ao Section508 e ADA. Os autores de websites podem utilizar um documento chamado VPAT (Voluntary Product Accessibility Template. Em tradução livre, a sigla representa "modelo de acessibilidade voluntária de produto"). Existem diferentes versões do documento para cada tipo de conformidade. Uma lista de exemplos para serem estudadas está disponível no link https://www.itic.org/policy/accessibility/vpat/.

Não existe um documento oficial desse tipo no Brasil, mas nem por isso devemos deixar de declarar a conformidade com padrões de acessibilidade. A WAI disponibiliza em seu site algumas ferramentas que permitem a geração de um relatório de conformidade para publicação em seu site.

Uma delas é a WCAG-EM Report Tool (https://www.w3.org/WAI/eval/report-tool/). Essa ferramenta ajuda a gerar um relatório de acordo com a WCAG-EM, *Website Accessibility Conformance Evaluation Methodology* (do inglês, em tradução livre, metodologia de avaliação de conformidade com acessibilidade de site). Essa ferramenta não realiza nenhuma verificação de acessibilidade, mas ajuda você a seguir as etapas da WCAG-EM (disponível em https://www.w3.org/TR/WCAG-EM/), para gerar um relatório estruturado a partir das informações fornecidas.

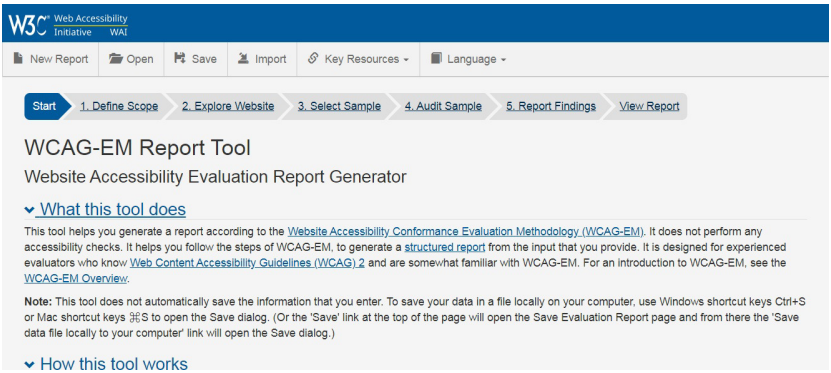

Web Accessibility Initiative WAI
New Report
Open
Save
Import
Key Resources
Language
Start
1. Define Scope
2. Explore Website
3. Select Sample
4. Audit Sample
5. Report Findings
View Report
WCAG-EM Report Tool
Website Accessibility Evaluation Report Generator
What this tool does
This tool helps you generate a report according to the Website Accessibility Conformance Evaluation Methodology (WCAG-EM). It does not perform any accessibility checks. It helps you follow the steps of WCAG-EM, to generate a structured report from the input that you provide. It is designed for experienced evaluators who know Web Content Accessibility Guidelines (WCAG) 2 and are somewhat familiar with WCAG-EM. For an introduction to WCAG-EM, see the WCAG-EM Overview.
Note: This tool does not automatically save the information that you enter. To save your data in a file locally on your computer, use Windows shortcut keys Ctrl+S or Mac shortcut keys ⌘S to open the Save dialog. (Or the 'Save' link at the top of the page will open the Save Evaluation Report page and from there the 'Save data file locally to your computer' link will open the Save dialog.)
How this tool works

Figura 12.1: Imagem da tela inicial do gerador de conformidade de acessibilidade WCAG-EM Report Tool

Seu uso é muito simples. São cinco etapas para gerar um relatório sobre a acessibilidade do seu site:

Definir o escopo

Esta área serve para declarar o que foi verificado no site, se todo o conteúdo exibido para desktops ou dispositivos móveis, qual a conformidade (WCAG 2 ou 2.1), o nível de conformidade (A, AA ou AAA) e até os agentes de usuário e tecnologias assistivas que são suportadas no site (como navegadores e leitores de tela).

Explorando o Website

Aqui serão descritas as tecnologias e padrões utilizados no desenvolvimento do site, como HTML5, CSS3, SVG, dentre outras. Também há espaço para descrever as funcionalidades do site e os tipos de página que podem ser encontradas, como páginas com outro layout ou estrutura de organização diferente de todo o conjunto.

Seleção de amostra representativa

Com base nas informações inseridas no passo anterior, você também pode declarar as funcionalidades essenciais do seu site e a variedade de páginas Web disponíveis. Também será necessário inserir algumas URLs de páginas como uma amostra que representa a maioria das páginas do site. Essa amostra ajuda a detectar erros em páginas com a mesma estrutura.

Auditando a amostra selecionada

Os links da amostra (inseridos no passo anterior) serão a base para o preenchimento deste relatório. Essa é a parte mais longa da geração do relatório onde serão inseridas todas as verificações feitas na página, com base no documento WCAG. Todos os princípios, diretrizes e critérios de sucesso devem ser verificados, e declarados como "não verificado", "passou", "falhou", "não presente" e "não posso dizer".

Relatando os resultados da avaliação

Essa última etapa permite que sejam inseridos os dados executivos para o relatório, como nome, nome do avaliador e data. Nessa parte também é possível fazer um resumo executivo de todo o relatório, onde é possível declarar de forma simples e rápida as observações principais encontradas na avaliação e uma avaliação geral da acessibilidade do site.

Ao final de todo o preenchimento, uma página com o resultado do relatório é exibida. Além de exibir o relatório na página, a ferramenta permite o download dos arquivos HTML, CSS e JSON do relatório.

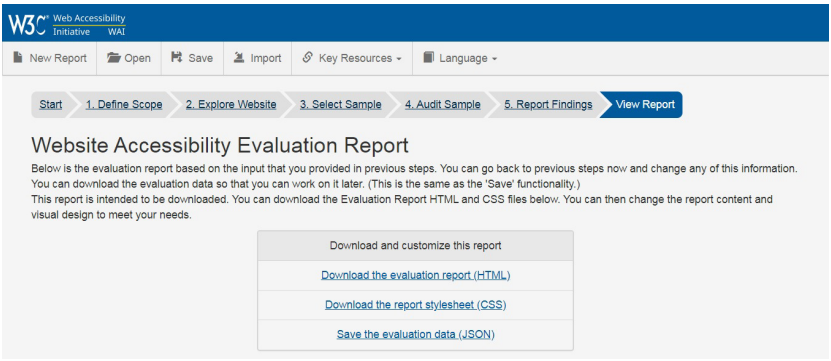

Web Accessibility Initiative WAI
New Report
Open
Save
Import
Key Resources
Language
Start
1. Define Scope
2. Explore Website
3. Select Sample
4. Audit Sample
5. Report Findings
View Report
Website Accessibility Evaluation Report
Below is the evaluation report based on the input that you provided in previous steps. You can go back to previous steps now and change any of this information.
You can download the evaluation data so that you can work on it later. (This is the same as the 'Save' functionality.)
This report is intended to be downloaded. You can download the Evaluation Report HTML and CSS files below. You can then change the report content and visual design to meet your needs.
Download and customize this report
Download the evaluation report (HTML)
Download the report stylesheet (CSS)
Save the evaluation data (JSON)

Figura 12.2: Imagem da tela final do gerador de conformidade de acessibilidade WCAG-EM Report Tool exibindo as opções de download do relatório

A segunda ferramenta também gera uma declaração de conformidade de forma mais simples, mas também com base em informações inseridas pelo usuário. O processo é o mesmo: os dados inseridos geram um documento de declaração de conformidade. O formulário está disponível no link: https://w3.org/WAI/planning/statements/generator/

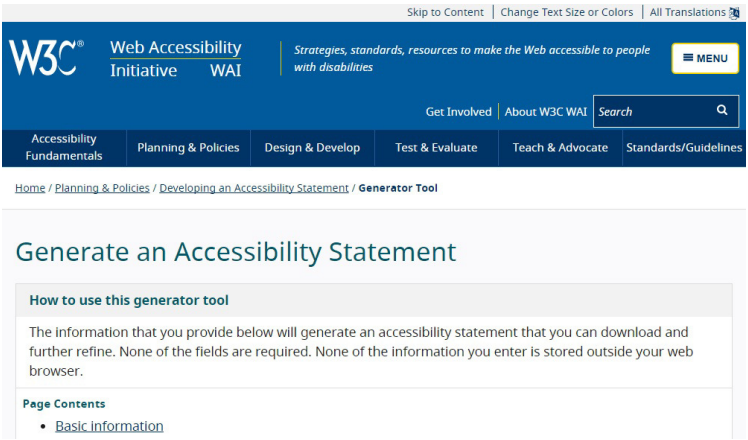

Skip to Content
Change Text Size or Colors
All Translations
Web Accessibility Initiative WAI
Strategies, standards, resources to make the Web accessible to people with disabilities
MENU
Get Involved
About W3C WAI
Search
Accessibility Fundamentals
Planning & Policies
Design & Develop
Test & Evaluate
Teach & Advocate
Standards/Guidelines
Home / Planning & Policies / Developing an Accessibility Statement / Generator Tool
Generate an Accessibility Statement
How to use this generator tool
The information that you provide below will generate an accessibility statement that you can download and further refine. None of the fields are required. None of the information you enter is stored outside your web browser.
Page Contents
Basic information

Figura 12.3: Imagem da tela inicial do gerador de conformidade de acessibilidade da WAI

Esse sistema pede diversas informações, separadas da seguinte forma:

Informações básicas

Pede informações referentes ao contato, endereço e nome do website até o nível de aderência com os padrões do W3C (WCAG 2.1, 2.0 ou outro). Aqui é possível escolher a declaração se o site está completamente de acordo com as WCAG até se não está em conformidade.

Esforços

Espaço destinado aos esforços que a organização teve para tornar o website acessível, como a criação de políticas ou a aplicação de acessibilidade em requisitos de qualidade.

Informações técnicas

Descrição de cenários onde pode haver barreiras de acesso e a combinação de recursos (como navegadores e tecnologia assistiva) mais compatível com o ambiente da aplicação no contexto da acessibilidade, além da declaração de toda tecnologia utilizada (HTML, CSS, JavaScript, ARIA etc.)

Aprovação e processo de conformidade

Lista o nome dos departamentos ou pessoas responsáveis pela aprovação de conformidade da acessibilidade da aplicação.

12.2 FAZENDO SUA PRÓPRIA DECLARAÇÃO DE CONFORMIDADE

No Brasil, essa prática não é comum e muito menos obrigatória (na verdade, existe uma lei que obriga a exibição de um "selo" na página). Você não precisa ter um gerador de conformidade para criar seu próprio documento. A parte mais importante desse processo é descrever a conformidade com padrões de acessibilidade, as barreiras de acesso que foram eliminadas e a data da verificação de conformidade.

Descreva ainda onde existe conteúdo que é de responsabilidade de terceiros, como um vídeo de um canal do qual você não tem a possibilidade de intervir ou em páginas de outros sites que foram referenciadas.

Além disso, a parte mais importante do processo é ter um canal aberto para receber comentários sobre a acessibilidade da aplicação. Lembre-se de que podem existir situações ou deficiências não contempladas e que muitas vezes não sabemos quais são. Manter esse canal aberto é uma ótima forma de saber o que o seu usuário precisa. Isso é importante pois é por ele que o usuário pode se comunicar de forma amigável; caso ele não consiga entrar em contato ou não seja respondido, ele pode acionar o dono do site judicialmente.

12.3 SELO DE ACESSIBILIDADE

Algumas instituições oferecem um selo de acessibilidade caso o site esteja em conformidade com diretrizes técnicas específicas. A Prefeitura de São Paulo oferece esse selo para sites que estejam em

conformidade com o padrão eMag e que passem por uma avaliação humana de especialistas. O selo informa os critérios utilizados e onde não se aplica a avaliação de acessibilidade.

Em outros países existem iniciativas simulares, que fazem uso de relatórios de conformidade somados à verificação automática de páginas. O selo também é uma forma de atestar que a aplicação está em conformidade com os padrões de acessibilidade.

12.4 ENCONTREI PROBLEMAS EM UM SITE. O QUE FAÇO?

Algumas instituições podem ajudar pessoas que encontraram barreiras de acesso em páginas. Um ótimo exemplo é a aplicação desenvolvida pelo Movimento Web para Todos, que recebe notificações de usuários contando sua experiência, seja ela boa ou ruim (https://mwpt.com.br/mobilizacao/).

Figura 12.4: Imagem da tela da aplicação do MWPT para que o usuário conte sua experiência relacionada à acessibilidade na Web

Isso ajuda não só o usuário a fazer um registro da acessibilidade da aplicação como as empresas a detectarem barreiras que elas podem nunca ter percebido. O Movimento ainda entra em contato com as instituições para ajudar a solucionar os problemas de acessibilidade. A aplicação cria um repositório de experiências que ajuda todos os interessados em acessibilidade, de usuários a empresas.

Caso o usuário não consiga solucionar seu problema com a ajuda do Movimento, existem outras alternativas para exigir acessibilidade de uma aplicação.

12.5 QUESTÕES LEGAIS

A Lei Nº 13.146, de 6 de julho de 2015 (conhecida com LBI, ou Lei Brasileira de inclusão) foi um grande avanço na acessibilidade na Web, já que a lei anterior dizia que os sites deveriam apenas ser acessíveis para pessoas com deficiência visual e não orientava como fazer isso. A LBI é mais específica. Seu artigo 63 diz o seguinte:

> Art. 63. É obrigatória a acessibilidade nos sítios da internet mantidos por empresas com sede ou representação comercial no País ou por órgãos de governo, para uso da pessoa com deficiência, garantindo-lhe acesso às informações disponíveis, conforme as melhores práticas e diretrizes de acessibilidade adotadas internacionalmente.

O fato de incluir em seu texto as "melhores práticas e diretrizes de acessibilidade adotadas internacionalmente" já é um grande avanço. Apesar de não dizer explicitamente como isso deve ser feito, as diretrizes internacionais adotadas internacionalmente são as WCAG.

A lei também fala sobre os sites exibirem um "selo de acessibilidade":

> § 1º Os sítios devem conter símbolo de acessibilidade em destaque.

Aqui existe um problema, pois não diz nem como deve ser esse selo nem a orientação de conformidade. Apenas colocar um selo no site não diz muito sobre o que deve ser feito, e nem como deve ser esse selo.

O fato é que, se uma pessoa com deficiência não conseguir acessar um site/serviço/aplicação devido à falta de acessibilidade, ela pode acionar a justiça para ter seus direitos preservados. O Ministério Público pode receber denúncias de cidadãos que tiveram seu direito de acesso negado e tomar as medidas cabíveis, que pode gerar um Termo de Ajuste de Conduta para exigir as correções necessárias para garantir o direito do cidadão.

O termo de conformidade não é uma garantia de que o site não pode ser acionado judicialmente por falta de acessibilidade. Não vou abordar a questão do uso de um termo no caso de má fé, publicando um termo em um site com barreiras de acesso, mas

sites podem sofrer atualizações que podem criar barreiras para os usuários. Por isso, manter um canal aberto com o usuário é fundamental.

12.6 DECLARAÇÃO NÃO GARANTE A ACESSIBILIDADE COMPLETA

Todas as formas de atestar a acessibilidade descritas aqui estão sujeitas a ter barreiras de acesso. O fato de ter uma declaração de conformidade ou selo é uma forma de apresentar ao público a preocupação com a acessibilidade, mas mesmo assim os usuários podem encontrar barreiras ao navegar, seja devido a uma atualização que adicionou uma barreira ou devido a uma barreira não prevista no processo de desenvolvimento.

O mais importante é manter um canal aberto com o usuário. As declarações e selos mostram o quanto o site está empenhado na acessibilidade. Deixe sempre uma forma de contato, seja por e-mail, telefone ou outra forma, para que os usuários possam reportar barreiras. Isso é importante pois os usuários podem ajudar a solucionar esses problemas de acessibilidade. Caso esse canal não esteja disponível o usuário pode partir para as vias legais. E isso envolve inclusive a participação do Ministério Público.

12.7 MINISTÉRIO PÚBLICO

Do lado do cidadão que encontra barreiras de acesso está o Ministério Público, que tem como principal função defender os interesses sociais e individuais do cidadão. Sendo assim, ele pode ser acionado caso um usuário tenha uma barreira de acesso em um

serviço e que seja comprovada o impedimento de acesso.

A Cartilha de Acessibilidade na Web do W3C Brasil (http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-II.html) ilustra muito bem as situações que envolvem o Ministério Público, mas vai além. Ela recomenda que o usuário, antes de acionar judicialmente um site/instituição, procure solucionar isso diretamente com o responsável pelo sistema. Sendo assim, uma pessoa que encontrar barreiras de acesso em um site deve:

1. **Procurar o responsável pelo site**: encontrar no site uma forma de contato para explicar as barreiras encontradas.
2. **Explique detalhadamente o problema**: liste todas as dificuldades encontradas. De preferência com links e arquivos que comprovem a barreira. Indique os detalhes técnicos, como sistema operacional, navegador e tecnologia assistiva utilizada.
3. **Indique fontes de informação**: você pode enviar links com orientações e diretrizes de acessibilidade que podem ajudar os responsáveis a identificar as barreiras.
4. **Solicite uma resposta**: peça que os responsáveis avisem quando o problema for solucionado.
5. **Esteja a disposição para ajudar**: não basta exigir que a aplicação elimine barreiras de acesso. Participe do processo e esteja à disposição para ajudar!

Esses passos são importantes, pois sem eles o Ministério Público não tem informações suficientes para ajudar. O contato com os responsáveis do site pode ser suficiente para a eliminação de barreiras mas, se isso não for possível, toda a comunicação

estabelecida pode fazer parte do material para ser encaminhado ao Ministério Público, que pode solicitar mais documentos antes de tomar uma ação. Essas ações podem ir desde uma notificação à empresa até a assinatura de um TAC (Termo de Ajuste de Conduta), documento que serve como um compromisso da empresa para realizar as correções necessárias.

Se nada disso funcionar, o Ministério Público pode levar o caso à justiça. As informações detalhadas sobre como o Ministério Público atua nos casos de acessibilidade na Web estão disponíveis no site do W3C Brasil (http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-II.html#capitulo3).

12.8 MUITO ALÉM DE QUESTÕES LEGAIS

Acessibilidade na Web não deveria ser contemplada apenas considerando o cumprimento de leis e questões legais. Ela deveria ser considerada de forma espontânea no desenvolvimento de qualquer aplicação. Ela permite que um público muito maior tenha acesso a aplicação e permite que todos, independente de alguma deficiência não tenham dificuldades em navegar.

Acesso à informação e serviços online é um direito do cidadão. Por isso, ele tem a justiça e órgãos públicos em sua defesa. Porém, pessoas fazem parte de todo o processo e empatia é algo fundamental para garantir a acessibilidade. Processos mais humanos aproximam as pessoas. Faça parte desse processo de transformação e não espere por ações judiciais para começar a pensar na acessibilidade digital.

Capítulo 13

A WEB QUE QUEREMOS - DE TODOS E PARA TODOS

Nos últimos anos temos acompanhado o crescimento da preocupação com os caminhos que a internet está trilhando. Na verdade, os caminhos que estão sendo trilhados para que a internet circule. São iniciativas, principalmente de grandes empresas, que tornam a rede um conjunto de pequenas bolhas e que faz com que não sejamos expostos a todo tipo de conteúdo. Ou pior - que sejamos expostos a conteúdos selecionados.

Esse é o momento de defendermos a Web dos riscos que ela vem correndo. Seu inventor, Tim Berners-Lee, vem propondo iniciativas para que a Web volte à sua essência e que o cidadão esteja no controle, e não grandes empresas. Tim vem publicando diversas cartas falando da sua preocupação com a Web, desde aquela resposta à Wired sobre "A morte da Web" (https://www.scientificamerican.com/article/long-live-the-web/) na revista Scientific American até as que ele vem publicando na Web Foundation (https://webfoundation.org/2019/03/web-birthday-30/) sobre um contrato para a Web, onde empresas, governos e pessoas assumem compromissos para garantir a evolução da Web.

Mas o que isso tem a ver com acessibilidade? Tudo!

Quando falamos de uma Web que queremos, não é somente uma que garanta nossa segurança e privacidade na rede. Queremos uma Web que não tenha barreiras para as pessoas. Essa Web que não discrimina pessoas por gênero, raça ou classe social também deve ser construída pensando nas pessoas com deficiência.

As barreiras técnicas talvez sejam as mais simples de serem transpostas, pois existem documentos, orientações e boas práticas para eliminar barreiras de acesso. O problema está na barreira comportamental.

Essa barreira comportamental é a mesma que nos mantém indiferentes a casos de racismo na rede, por exemplo. O fato de não existir uma ferramenta ou um "guia de boas práticas" para evitar comportamentos de segregação ou hostis torna o abismo entre as pessoas muito maior. O simples fato de estar do outro lado de uma tela nos distancia das pessoas. E isso tem um enorme impacto para a acessibilidade.

Quando não levamos em consideração o acesso de pessoas com deficiência, estamos tirando o direito de uma pessoa de navegar, interagir ou consumir produtos e serviços na rede. Isso pode ser refletido nas mais diversas situações: uma pessoa cega que não consegue submeter um currículo para uma vaga de emprego, uma pessoa com movimentos limitados que não consegue encontrar um endereço em um mapa, uma pessoa idosa que não consegue ler e compreender os contratos do seu plano de saúde porque o texto é enorme e as letras são muito pequenas, e por aí vai.

Isso acontece porque em algum momento do desenvolvimento

da aplicação não consideramos as diferenças. Não pensamos que as pessoas acessam e navegam de formas distintas e isso tem uma relação muito grande com a forma como estamos incluindo a diversidade em nossas aplicações.

O uso de smartphones colocou novas situações de uso, que começaram a ser consideradas nos estudos de usabilidade. Navegar com uma mão só, tocar em botões pequenos e ler em telas pequenas são alguns exemplos. Mas parte desses profissionais de UX já sabe que além das diferenças de dispositivos existe a diferença entre as pessoas. Será que todos os desenvolvedores de aplicação consideram, por exemplo, uma pessoa sem braços quando pensam na interface de um dispositivo?

Dentro do próprio W3C já existem iniciativas no sentido de trazer mais diversidade para o consórcio. Essa diversidade é muito rica para que tenhamos mais empatia com as diferenças e consigamos levar em conta um maior número de pessoas. Dessas iniciativas, gostaria de destacar as seguintes:

- **Education and Outreach Working Group (EOWG)** - https://www.w3.org/WAI/about/groups/eowg/. Grupo dedicado a desenvolver estratégias e recursos para promover a conscientização, compreensão e implementação da acessibilidade na Web. Este é o grupo que desenvolve os vídeos "Web Accessibility Perspectives" (https://www.w3.org/WAI/perspective-videos/), que mostram como a acessibilidade é importante para todos;
- **Inclusion and Diversity Community Group** - https://www.w3.org/community/idcg/. Este Community Group tem como objetivo aumentar a presença de grupos

sub-representados no W3C e fortalecer a cultura do W3C, apoiando a diversidade;

- **Positive Work Enviroment Community Group** - https://www.w3.org/community/pwe/. Grupo destinado a discutir e aproximar pessoas das mais diversas culturas, idiomas, nacionalidades e experiências.

A Web que queremos depende de nós. Depende da nossa participação e posicionamento em questões que envolvem não só a parte da governança da internet como da inclusão de todas as pessoas. Costumo dizer que se não pensarmos a acessibilidade em aplicações desenvolvidas hoje, talvez daqui a vinte ou trinta anos teremos dificuldade em utilizar essa aplicação devido ao envelhecimento.

Acessibilidade não é filantropia. É respeito pelas pessoas e por um mundo efetivamente justo em oportunidades para todos. Em apresentações costumo usar uma frase adaptada da arquiteta Thais Frota para o contexto da Web da seguinte forma:

> Se o seu site não está pronto para receber todas as pessoas, o seu site é deficiente!

Essa frase tem uma grande mensagem que é mostrar que as barreiras de acesso são colocadas por nós. Elas não "aparecem" do nada para dificultar o acesso. As diferenças existem para serem respeitadas e é isso que devemos buscar na Web que queremos.

Participe! Colabore! Construa uma Web acessível para todos!

Capítulo 14

CONSIDERAÇÕES FINAIS

Chegamos ao final da publicação. Espero que tenham gostado deste livro, pois escrevi com muito carinho e cuidado. Parabéns a você que se interessou por esse tema, seja para conhecer ou se aprofundar em acessibilidade na Web. Fico feliz em saber que pude contribuir para que a Web seja um ambiente pelo menos um pouco mais inclusivo.

Ao término deste livro espero que você tenha identificado questões importantes com relação à acessibilidade na Web:

- **Uma Web acessível beneficia todas as pessoas**. Conhecemos como as pessoas com deficiência navegam e os tipos de tecnologia assistiva que utilizam, mas também pudemos compreender como a acessibilidade beneficia pessoas idosas, pessoas com deficiência temporária e outros públicos.
- **Eliminar barreiras de acesso é mais simples do que parece**. Pudemos perceber como pequenas intervenções no código de um site podem quebrar grandes barreiras na Web. São práticas que podem ser adicionadas no dia a dia do desenvolvimento.
- **Acessibilidade não é filantropia. É garantia de direitos iguais**. Aprendemos que existem leis, direitos e instituições

de defesa dos direitos da pessoa com deficiência. Desrespeitar essas leis e direitos podem ter implicações legais.

O objetivo deste livro não é terminar por aqui. Existem duas coisas que eu gostaria de deixar como uma "lição de casa", após ler todo esse material:

Use este livro como referência sempre que precisar

Escrevi este livro pensando em como eu gostaria de ter estudado para conhecer a acessibilidade na Web. Quando comecei fui pelo caminho mais longo e doloroso, que foi destrinchar as diretrizes do W3C. Este material é para que as pessoas possam compreender de forma muito mais simples o que essas diretrizes representam e que sejam uma fonte de referências durante estudos ou mesmo sobre um projeto.

Participe de iniciativas em favor da acessibilidade

Existem diversos grupos, tanto nacionais como internacionais, que atuam em favor da Web acessível. Faça parte das comunidades como as listas de discussão públicas do W3C internacional, do W3C Brasil, do Movimento Web para Todos, Web sem Barreiras e outras iniciativas.

Defenda a Web acessível

Precisamos de mais pessoas engajadas não só em desenvolver sites acessíveis como também para transmitir a mensagem da Web sem barreiras para todas as pessoas. Leve esse tema para reuniões, palestras e eventos dos quais participa. Não deixe que esse tema

seja esquecido.

Para finalizar, quero deixar uma reflexão do saudoso MAQ (o Marco Antônio de Queiroz, um dos pioneiros da acessibilidade digital no Brasil), que dizia seu sonho era que a acessibilidade na Web não fosse mais necessária, já que todos colocariam a acessibilidade como parte da rotina e boas práticas com o desenvolvimento. Compartilho desse sentimento, e espero que um dia este livro se torne desnecessário.

Estou à disposição para tirar dúvidas e orientar sobre questões relacionadas à acessibilidade digital. Fiquem à vontade para entrar em contato comigo quando quiserem pelos canais disponíveis no meu site (http://www.reinaldoferraz.com.br/sobre/).

Até a próxima!

Capítulo 15

REFERÊNCIAS

Como este livro é feito para consumo contínuo, nada mais justo do que listar todos os links externos desta publicação em um lugar só, organizados por tipo de referência. Todos os links foram verificados em janeiro de 2020.

15.1 PADRÕES E DOCUMENTOS DO W3C

- WCAG (versão mais atual):

https://www.w3.org/TR/WCAG/

- WCAG 1.0:

https://www.w3.org/TR/WAI-WEBCONTENT/

- WCAG 2.0:

https://www.w3.org/TR/WCAG20/

- WCAG 2.0 padrão ISO:

https://www.iso.org/standard/58625.html

- WCAG 2.0 em português (tradução autorizada):

https://www.w3.org/Translations/WCAG20-pt-br/

- WCAG 2.1:

https://www.w3.org/TR/WCAG21/

- WCAG 2.1 em português (tradução livre):

http://www.w3c.br/traducoes/wcag/wcag21-pt-BR/

- WCAG-EM:

https://www.w3.org/TR/WCAG-EM/

- WAI-ARIA 1.1:

https://www.w3.org/TR/wai-aria-1.1/

- WAI-ARIA 1.1 - Definition of Roles:

https://www.w3.org/TR/wai-aria-1.1/#role_definitions

- WAI-ARIA 1.1 - States and Properties:

https://www.w3.org/WAI/PF/aria-1.1/states_and_properties

- ATAG 1.0:

https://www.w3.org/TR/ATAG10/

- UAAG:

https://www.w3.org/TR/UAAG/

- TTML 2.0:

https://www.w3.org/TR/2018/REC-ttml2-20181108/

- HTML 5.2 - Elements:

https://www.w3.org/TR/html52/dom.html#elements

- Evaluation and Report Language (EARL) Overview:

https://www.w3.org/WAI/standards-guidelines/earl/

- Education and Outreach Working Group:

https://www.w3.org/WAI/about/groups/eowg/

- Web Accessibility Perspectives videos:

https://www.w3.org/WAI/perspective-videos/

- Introduction to Web Accessibility:

https://www.w3.org/WAI/fundamentals/accessibility-intro/

- Diverse Abilities and Barriers:

https://www.w3.org/WAI/people-use-web/abilities-barriers/

- Inclusion and Diversity Community Group:

https://www.w3.org/community/idcg/

- Positive Work Environment Community Group:

https://www.w3.org/community/pwe/

- Generate an Accessibility Statement:

https://w3.org/WAI/planning/statements/generator/

- Mobile Accessibility at W3C:

https://www.w3.org/WAI/standards-guidelines/mobile/

- Accessibility Conformance Testing (ACT) Rules Format 1.0:

https://www.w3.org/TR/act-rules-format/

- World Wide Web Consortium Launches International Program Office for Web Accessibility Initiative:

https://www.w3.org/Press/IPO-announce

- Cartilha de Acessibilidade na Web, W3C Brasil - Fascículo 1:

http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html

- Cartilha de Acessibilidade na Web, W3C Brasil - Fascículo 2:

http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-II.html

- Cartilha de Acessibilidade na Web, W3C Brasil - Fascículo 3:

http://www.w3c.br/Materiais/materiais/cartilha-w3cbr-acessibilidade-web-fasciculo-III.html

- Techniques for the The Cognitive and Learning Disabilities

Accessibility Task Force (COGA):

https://w3c.github.io/coga/techniques/

15.2 ARTIGOS E BOAS PRÁTICAS

- Sites de e-commerce não estão preparados para receber pessoas com deficiência:

https://mwpt.com.br/sites-de-e-commerce-nao-estao-preparados-para-receber-pessoas-com-deficiencia/

- Don't Use Tabindex Greater than 0:

https://adrianroselli.com/2014/11/dont-use-tabindex-greater-than-0.html

- Google deixa de mostrar nas buscas sites não responsivos:

https://exame.abril.com.br/tecnologia/google-deixa-de-mostrar-nas-buscas-sites-nao-responsivos/

- How to design visual learning resources for neurodiverse students:

https://blog.fullfabric.com/how-to-design-visual-learning-resources-curriculum-for-neurodiverse-audience-autism-autistic

- Long Live the Web: A Call for Continued Open Standards and Neutrality:

https://www.scientificamerican.com/article/long-live-the-web/

- 30 years on, what's next ForTheWeb?

https://webfoundation.org/2019/03/web-birthday-30/

- Explorando atributos web relacionados à acessibilidade em imagens e seu impacto sobre a indexação por ferramentas de busca:

http://ceweb.br/media/docs/publicacoes/19/acessibilidade-explorando-atributos-web.pdf

- Acessibilidade, SEO e SVG:

http://www.reinaldoferraz.com.br/acessibilidade-seo-e-svg/

- Acessibilidade, SEO e SVG (artigo completo):

http://ceweb.br/media/docs/publicacoes/19/acessibilidade-svg-seo-Cewebbr-ReinaldoFerraz.pdf

- How the internet still fails disabled people:

https://www.theguardian.com/technology/2015/jun/29/disabled-people-internet-extra-costs-commission-scope

- A History of Accessibility at IBM:

https://www.afb.org/afbpress/pubnew.asp?DocID=aw050207

- Webstandards.org:

https://www.webstandards.org/

- Projeto GAIA:

http://talitapagani.com/gaia/sobre/

15.3 LEIS E REGULAMENTAÇÕES

- Decreto Nº 6.949, de 25 de agosto de 2009 que Promulga a Convenção Internacional sobre os Direitos das Pessoas com Deficiência:

http://www.planalto.gov.br/ccivil_03/ato2007-2010/2009/decreto/d6949.htm

- Lei Nº 13.146, de 6 de julho DE 2015 que Institui a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência):

http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm

- eMAG - Modelo de Acessibilidade em Governo Eletrônico:

http://emag.governoeletronico.gov.br/

- VPAT:

https://www.itic.org/policy/accessibility/vpat/

- Section 508:

https://www.section508.gov/

- Section 508 - Create Accessible Software & Websites:

https://www.section508.gov/create/software-websites/

- Americans with Disabilities Act:

https://www.ada.gov/

- Accessibility of State and Local Government Websites to People with Disabilities:

https://www.ada.gov/websites2.htm

15.4 TECNOLOGIA ASSISTIVA E DE APOIO

- Leitor de tela JAWS (para Windows):

http://www.freedomscientific.com/products/software/jaws/

- Documentação de teclas de atalho do JAWS:

https://support.freedomscientific.com/Content/Documents/Manuals/JAWS/Keystrokes.pdf

- Leitor de tela NVDA (para Windows):

https://www.nvaccess.org/download/

- Documentação do NVDA:

https://www.nvaccess.org/files/nvda/documentation/userGuide.html?#BrowseMode/

- Leitor de tela ORCA (para Linux):

https://wiki.gnome.org/Projects/Orca/

- Documentação do ORCA:

https://help.gnome.org/users/orca/stable/commands.html

- Leitor de tela Chromevox (Plug-in para Google Chrome):

https://chrome.google.com/webstore/detail/chromevox/kgejglhpjiefppelpmljglcjbhoiplfn?hl=pt-BR

- Documentação do Chromevox:

https://support.google.com/chromebook/answer/7031755?hl=pt-BR

- Leitor de tela VoiceOver (para iOS):

https://www.apple.com/br/accessibility/iphone/vision/

- Documentação do VoiceOver:

https://help.apple.com/iphone/10/?lang=pt-br#/iph75e97af9b

- Leitor de tela Talkback (Para Android):

https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback&hl=pt

- Documentação do Talkback:

https://support.google.com/accessibility/android/answer/6151827

- Ampliador de tela Lente Pré (para Windows):

http://intervox.nce.ufrj.br/dosvox/dosvox.html

- Ampliador de tela Magic (para Windows):

http://www.freedomscientific.com/products/software/magic/

- High Contrast (Plugin para Google Chrome):

https://chrome.google.com/webstore/detail/high-contrast/djcfdncoelnlbldjfhinnjlhdjlikmph?hl=pt-BR

- Text Legibility (Plugin para Google Chrome):

https://addons.mozilla.org/pt-BR/firefox/addon/text-legibility/?src=search

- Tradutor de LIBRAS Handtalk:

https://www.handtalk.me/

- Tradutor de LIBRAS VLibras:

http://www.vlibras.gov.br/

- Transcrição de áudio automático - WebCaptioner:

https://webcaptioner.com/

- Configuração de navegação para dispositivos Apple:

https://support.apple.com/en-us/HT201370

- EVA Facial Mouse:

https://play.google.com/store/apps/details?id=com.crea_si.eviacam.service

- Documentação da SIRI:

https://www.apple.com/br/siri/

- Pesquisa por voz do Google:

https://support.google.com/websearch/answer/2940021?

co=GENIE.Platform%3DAndroid&hl=en&oco=0

- WebHelp (Plugin para Google Chrome):

https://chrome.google.com/webstore/detail/webhelp/pjnhjelpkdoihfjeeemmahpdbmgliboo?hl=pt-BR

- Dyslexia Formatter (Plugin para Google Chrome):

https://chrome.google.com/webstore/detail/dyslexia-formatter/kggkghfhlppjclojgphbploiaipgogoc?hl=pt-BR

15.5 FERRAMENTAS

- HTML5 Accessibility:

https://www.html5accessibility.com/

- Contrast Checker:

https://webaim.org/resources/contrastchecker/

- Color Contrast Analyzer:

https://chrome.google.com/webstore/detail/color-contrast-analyzer/dagdlcijhfbmgkjokkjicnnfimlebcll

- WCAG Contrast checker:

https://addons.mozilla.org/pt-BR/firefox/addon/wcag-contrast-checker/

- WCAG-EM Report Tool:

https://www.w3.org/WAI/eval/report-tool/

- Colorblind Web Page Filter:

https://www.toptal.com/designers/colorfilter/

- Validador de markup do W3C (online):

https://validator.w3.org/

- Validador de markup do W3C (para instalação):

https://validator.w3.org/docs/install.html

- Validador de Feed RSS do W3C:

https://validator.w3.org/feed/

- Verificador de links do W3C:

https://validator.w3.org/checklink

- Validador de CSS do W3C:

http://jigsaw.w3.org/css-validator/

- Lista de verificadores de acessibilidade do W3C:

https://www.w3.org/WAI/ER/tools/

- Verificador de acessibilidade WAVE:

http://wave.webaim.org/

- Verificador de acessibilidade HTML_Codesniffer:

https://squizlabs.github.io/HTML_CodeSniffer/

- Verificador de acessibilidade AChecker:

https://achecker.ca/

- Verificador de acessibilidade Mauve:

https://mauve.isti.cnr.it/

- Verificador de acessibilidade Access Monitor Plus:

http://accessmonitor.acessibilidade.gov.pt/amp/

- Verificador de acessibilidade ASES (online):

http://asesweb.governoeletronico.gov.br/ases/

- Verificador de acessibilidade Access Monitor Plus:

http://accessmonitor.acessibilidade.gov.pt/amp/

- Verificador de acessibilidade ASES (para download):

https://softwarepublico.gov.br/social/ases

- Tabela de cores e contrastes do eMag:

http://emag.governoeletronico.gov.br/#r1.5

15.6 DOCUMENTAÇÃO DE FRAMEWORKS

- Days since last JavaScript framework:

https://dayssincelastjavascriptframework.com/

- Wordpress Theme Review Team:

https://make.wordpress.org/themes/handbook/review/accessibility/

- Wordpress Theme Directory:

https://wordpress.org/themes/tags/accessibility-ready/

- Wordpress Accessibility:

https://en.support.wordpress.com/accessibility/

- Wordpress Plugin Tag: accessibility:

https://wordpress.org/plugins/tags/accessibility/

- Joomla! Extensions Directory:

https://extensions.joomla.org/tags/accessibility/

- WAI-ARIA Roles in Accessible Admin Template:

https://community.joomla.org/blogs/community/wai-aria-roles-in-accessible-admin-template.html

- Plone Accessibility:

https://plone.org/accessibility-info/

- Static AST checker for a11y rules on JSX elements:

https://github.com/evcohen/eslint-plugin-jsx-a11y

- Identifies accessibility issues in your React.js elements:

https://github.com/reactjs/react-a11y

- A fully accessible React modal built according WAI-ARIA Authoring Practices:

https://github.com/davidtheclark/react-aria-modal

- Vue A11Y:

https://github.com/vue-a11y/

- Accessibility in Angular:

https://angular.io/guide/accessibility/

- Angular/components:

https://github.com/angular/components/tree/master/src/cdk/

- Angular Accessibility:

https://material.angular.io/cdk/a11y/overview/

- Ember - Intro to Accessibility:

https://guides.emberjs.com/release/reference/accessibility-guide/

- Ember accessibility and a11y tools:

https://blog.emberjs.com/2018/06/17/ember-accessibility-and-a11y-tools.html/

- JQuery - Tagged: accessibility:

https://plugins.jquery.com/tag/accessibility/

15.7 RECURSOS DIVERSOS

- Home of the first website:

http://info.cern.ch/

- A primeira foto da Web:

https://en.wikipedia.org/wiki/File:Les_Horribles_Cernettes_in_1992.jpg

- E-mail de 1993 sobre a inserção do elemento `img` :

http://1997.webhistory.org/www.lists/www-talk.1993q1/0182.html

- E-mail de 1995 sobre a inserção do atributo `alt` :

http://1997.webhistory.org/www.lists/www-html.1995q2/0128.html

- Censo IBGE 2010:

https://censo2010.ibge.gov.br/noticias-censo?id=3&idnoticia=2170&view=noticia

- Portal de Dados Cetic.br:

http://data.cetic.br/

- Screen Reader User Survey 7 Results:

https://webaim.org/projects/screenreadersurvey7/

- Pesquisa sobre o uso de leitores de tela - 2ª edição:

https://estudoinclusivo.com.br/

- Game Accessibility Guidelines:

http://gameaccessibilityguidelines.com/

- IANA - lista de idiomas:

https://www.iana.org/assignments/language-subtag-registry/language-subtag-registry

- Movimento Web para Todos:

https://mwpt.com.br/mobilizacao/

- Website pessoal - Reinaldo Ferraz:

http://www.reinaldoferraz.com.br/

- W3C Brasil:

https://w3c.br/

- Ceweb.br:

https://www.ceweb.br/

- Reinaldo Ferraz - Twitter:

https://twitter.com/reinaldoferraz/

- Reinaldo Ferraz - Instagram:

https://www.instagram.com/reinaldoferraz/

- Reinaldo Ferraz - Wikipedia:

https://pt.wikipedia.org/wiki/Reinaldo_Ferraz

- Reinaldo Ferraz - Linkedin:

https://www.linkedin.com/in/reinaldoferraz/